

VISIT...

LANZAROTE
Caliente.COM

Oliver Montenbruck
Thomas Pfleger

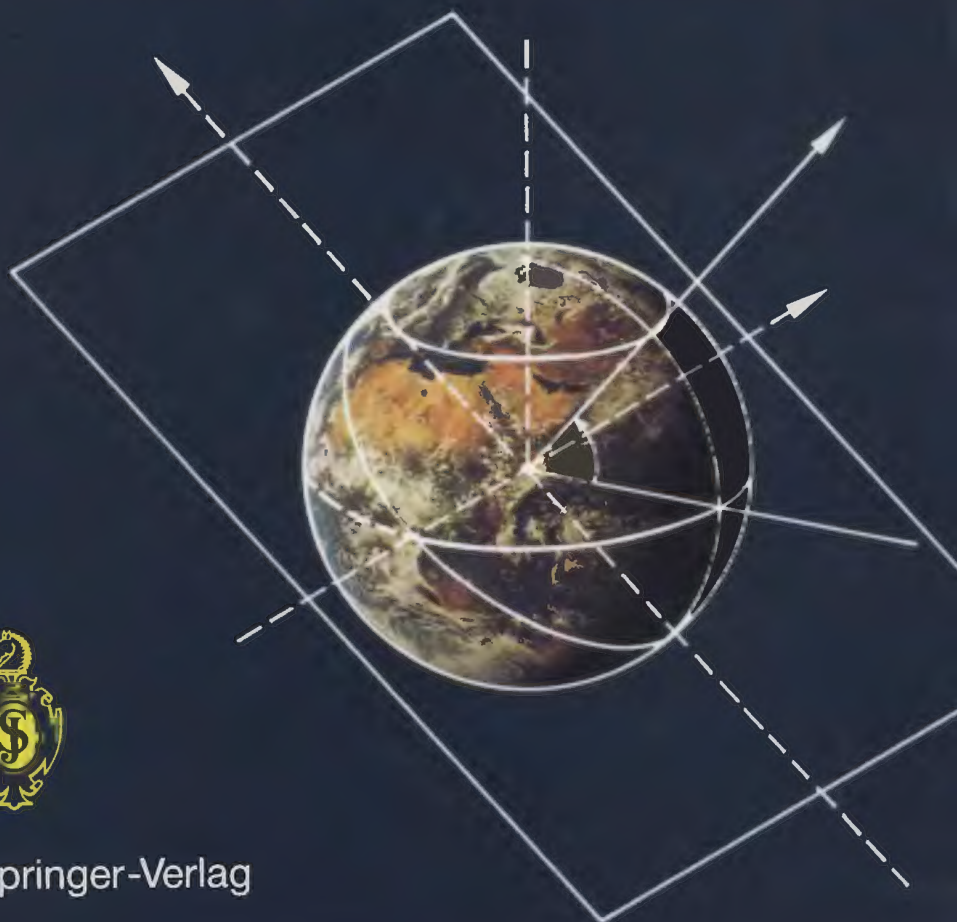


Second, Corrected
and Enlarged Edition

Astronomy on the Personal Computer



Springer-Verlag



Oliver Montenbruck Thomas Pfleger

Astronomy on the Personal Computer

Translated by Storm Dunlop

Second, Corrected and Enlarged Edition
With 45 Figures and a Floppy Disk

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Dr. rer. nat. Oliver Montenbruck

DLR-GSOC, D-82230 Weßling, Germany

Dipl.-Ing. Thomas Pfleger

Schloßstraße 56, D-53773 Hennef, Germany

Translator:

Dr. Storm Dunlop

140 Stocks Lane, East Wittering, Chichester
West Sussex PO20 8NT, United Kingdom

Title of the original German edition:

O. Montenbruck, T. Pfleger: *Astronomie mit dem Personal Computer*

Zweite, überarbeitete und stark erweiterte Auflage

© Springer-Verlag Berlin Heidelberg 1989, 1994

ISBN 3-540-57701-7 2. Auflage Springer-Verlag Berlin Heidelberg

ISBN 3-540-57700-9 2. Auflage Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-57700-9 2nd Edition Springer-Verlag New York Berlin Heidelberg

ISBN 3-540-52754-0 1. Auflage Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-52754-0 1st Edition Springer-Verlag New York Berlin Heidelberg

Library of Congress Cataloging-in-Publication Data. Montenbruck, Oliver, 1961-. [Astronomie mit dem Personal Computer. English] Astronomy on the personal computer / O. Montenbruck, T. Pfleger; translated by S. Dunlop. – 2nd corr. and enl. ed. p. cm. Includes bibliographical references and index. ISBN 3-540-57700-9 (Berlin: acid-free paper). – ISBN 0-387-57700-9 (New York: acid-free paper) 1. Astronomy—Data processing. 2. Microcomputers. I. Pfleger, T. (Thomas), 1964-. II. Title. QB51.3.E43M6613 1994 522'.85'536—dc20 94-28233

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1991, 1994

Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Please note: Before using the programs in this book, please consult the technical manuals provided by the manufacturer of the computer – and of any additional plug-in boards – to be used. The authors and the publisher accept no legal responsibility for any damage by improper use of the instructions and programs contained herein. Although these programs have been tested with extreme care, we can offer no formal guarantee that they will function correctly.

The programs on the enclosed diskette are under copyright-protection and may not be reproduced without written permission by Springer-Verlag. One copy of the programs may be made as a back-up, but all further copies offend copyright law.

Typesetting: Camera-ready copy from the authors

Cover design: Erich Kirchner, Heidelberg, Germany

SPIN: 10131861

55/3140 – 5 4 3 2 1 0 – Printed on acid-free paper

Foreword

It is said that a typical astronomer of the 19th century spent seven hours working at a desk for every hour spent at the telescope. That's how long the routine analysis of data took with pencil, paper, and logarithmic tables. Thus when Wilhelm Olbers discovered the minor planet Vesta in 1807 and gathered the necessary observations, his friend Gauss needed almost 10 hours to hand-calculate its orbit. That achievement astonished many less gifted astronomers of the time, who might have labored days to work out the orbit of a newfound comet.

How different things are today! Gauss's method of orbit determination, presented in Chap. 11 of this book, runs to completion on a home computer in a few seconds at most. The machine will issue its accurate results in less time than it takes to key in the observations.

In this book, a landmark in the youthful literature of astronomical computer algorithms, Oliver Montenbruck and Thomas Pfleger cover many topics of keen interest to the practical observer. For me its most remarkable feature is the library of interrelated program modules, all elegantly written in PASCAL. Anyone who has tried to create such modules in interpreted BASIC soon runs into trouble: too few letters for variable names, not enough significant digits, and so on. These PASCAL routines are invoked one after another in coordinate transformations and calendar conversions. As "building blocks" for writing programs they find application in all corners of astronomy, not just the specific topics treated. That is reason enough, in my view, for anyone not so equipped to go out and buy a PASCAL compiler at once.

Any active observer will welcome the power and generality of these programs. Chapter 10, for example, tells how to make personal predictions of all stars to be occulted by the Moon for a given spot on Earth. It's exciting to see a star wink out of sight or pop into view from behind the Moon's limb, but one has to know when and where to look. For many years, serious amateurs around the world have carefully timed such events, thereby gathering the raw data for detailed study of the Moon's limb profile and orbital motion. Astronomy magazines can alert their readers to just a few of the most spectacular occultations each year – yet they happen all the time.

Comets and minor planets offer many opportunities for amateur-professional cooperation as well. A large observatory can't easily shuffle its schedule to cope with the sudden appearance of a bright comet. Yet countless ama-

teurs already possess the telescopes, photographic skills, and enthusiasm to swing into action at a moment's notice. This book's chapters on measuring photographs, determining orbits, and calculating an ephemeris are just what is needed to participate in discovering and publicizing a new comet. Personal computers, especially when equipped with hard disks and math coprocessors, are also well suited for the specialized work of identifying and improving the orbits of minor planets – something that just a few years ago was the exclusive province of large mainframes.

Total solar eclipses hold a special fascination for astronomers, but accurate prediction has not always been easy. In October of 1780, an expedition from Harvard College sailed to Penobscot Bay on the coast of Maine and somehow managed to set up its telescopes 30 km outside the path of totality! Such a blunder is far less likely today, even if official predictions are not issued until a few months before an eclipse takes place. However, owners of this book are spared even that frustration; the eclipse program of Chap. 9 will serve quite well for detailed travel plans.

Indeed, the carefully crafted computer programs presented here go a long way toward freeing us from the need for an almanac or yearbook, the traditional “astronomer's bible”. While I still cherish such works on my bookshelf for their accurate planet positions, occultations, and eclipse calculations, I'm starting to look on them differently. Once they seemed great repositories of astronomical knowledge, full of data not obtainable elsewhere. More and more, they now serve as checks on the wonderfully accurate predictions I'm learning to make with my home computer.

Cambridge, MA
October 1990

Roger W. Sinnott

Preface to the Second Edition

Since the publication of the first edition of *Astronomy on the Personal Computer*, we have received numerous comments and suggestions. Together with the publishers' interest in a new edition, this prompted us to revise the book and to incorporate a wide range of improvements to the text.

The first important addition is a chapter on the calculation of perturbations. This shows how gravitational perturbations by the major planets may be incorporated into the calculation of ephemerides for minor planets and comets. The NUMINT program described here enables more accurate positions to be calculated. This will be of considerable assistance in searching for what are often extremely faint objects. This tool for calculating perturbations complements the chapters on the calculation of ephemerides, the determination of orbits, and astrometry.

The second additional chapter discusses the calculation of physical ephemerides for the major planets and the Sun, and thus fills a gap in the earlier edition's coverage. Amateurs now have at their disposal the means of both predicting and subsequently reducing interesting planetary observations.

Other changes mainly concern the calculation of rising and setting times for the planets, and of the local circumstances that apply to solar eclipses.

Finally, there is now a single version of the program diskette that is supplied with the book. Since the firm of Application Systems Heidelberg have introduced their Pure Pascal compiler for Atari ST/TT computers that is compatible with Borland's Turbo Pascal, there is no need for a special Atari version of the program diskette. The enclosed diskette may, therefore, be used without modifications with Turbo Pascal on IBM-compatible machines or with Pure Pascal on Atari computers. Details about the appropriate installation procedures may be found in the AAREADME.DOC file on the diskette.

We should like to thank Springer-Verlag for their helpful co-operation, and also Application Systems Heidelberg for their technical support.

Munich, July 1994

O. Montenbruck and T. Pfleger

Preface to the First Edition

Nowadays anyone who deals with astronomical computations, either as a hobby or as part of their job, inevitably turns to using a computer. This is particularly true now that personal computers have become firmly established as ubiquitous aids to living. Calculations that could not even be contemplated a short time ago are now available to a whole range of users, and at no farther remove than their desks.

Not only has the technical capacity of computers grown, but so has the need for powerful – i.e., fast and accurate – programs. So the wish to avoid conventional, astronomical yearbooks as much as possible is quite understandable.

We were therefore delighted to take up our publisher's suggestion that we explain the fundamental principles of spherical astronomy, ephemeris calculations, and celestial mechanics in the form of this book.

Astronomy on the Personal Computer offers readers who develop their own programs a comprehensive library of Pascal procedures for solving a whole range of individual steps that frequently occur in problems. This includes routines for common coordinate transformations, for time and calendar calculations, and for handling the two-body problem. Specific procedures allow the exact positions of the Sun, the Moon, and the planets to be calculated, taking mutual perturbations into account. Thanks to the widespread use of Pascal as a computing language, and by avoiding computer-specific commands, the programs may be employed on a wide range of modern computers from the PC to the largest mainframes. The large number of routines discussed should at least mean that few readers will have to 're-invent the wheel', and that they will therefore be free to concentrate on their own particular interests.

Each chapter of this book deals with a fairly restricted theme and ends with a complete main program. From simple questions, such as the determination of rising and setting times or the calculation of the positions of the planets, more complex themes are developed, such as the calculation of solar eclipses and stellar occultations. The programs for the astrometric reduction of photographs of star fields and for orbit determination enable users to derive orbital elements of comets or minor planets for themselves. Even readers without programming experience will be able to use the appropriate applications.

Sufficient details are given of the astronomical and mathematical grounds on which solutions of specific problems are based for readers to understand the programs presented. This knowledge will enable them to adapt any of the programs to their individual needs. This close link between theory and practice also enables us to explain what are sometimes quite complex aspects in a much easier fashion than the descriptions found in classical textbooks. To sum up, we hope that we have given readers a fundamental grounding in using computers for astronomy.

We should like to thank S. Dunlop for producing such an excellent translation and Dr. G. Wolschin and C.-D. Bachem of Springer-Verlag for their cordial cooperation and interest during the process of publishing this book. Our thanks are also due to all our friends and colleagues, who, with their ideas and advice, and their help in correcting the manuscript and in testing the programs, have played an important part in the success of this book.

Munich, August 1990

O. Montenbruck and T. Pfleger

Contents

1	Introduction	1
1.1	Some Examples	1
1.2	Astronomy and Computing	2
1.3	Programming Languages and Techniques	4
2	Coordinate Systems	7
2.1	Making a Start	7
2.2	Calendar and Julian Dates	11
2.3	Ecliptic and Equatorial Coordinates	13
2.4	Precession	17
2.5	Geocentric Coordinates and the Orbit of the Sun	23
2.6	The COCO Program	27
3	Calculation of Rising and Setting Times	35
3.1	The Observer's Horizon System	35
3.2	Sun and Moon	38
3.3	Sidereal Time and Hour Angle	40
3.4	Universal Time and Ephemeris Time	41
3.5	Parallax and Refraction	45
3.6	Rising and Setting Times	47
3.7	Quadratic Interpolation	49
3.8	The SUNSET Program	50
3.9	The PLANRISE Program	57
4	Cometary Orbits	59
4.1	Form and Orientation of the Orbit	59
4.2	Position in the Orbit	61
4.3	Mathematical Treatment of Kepler's Equation	65
4.4	Near-Parabolic Orbits	68
4.5	Gaussian Vectors	71
4.6	Light-Time	76
4.7	The COMET Program	77

5	Special Perturbations	83
5.1	Equation of Motion	84
5.2	Planetary Coordinates	87
5.3	Numerical Integration	89
5.4	Osculating Elements	95
5.5	The NUMINT Program	98
6	Planetary Orbits	107
6.1	Series Expansion of the Kepler Problem	108
6.2	Perturbation Terms	111
6.3	Numerical Treatment of the Series Expansions	114
6.4	Apparent and Astrometric Coordinates	118
6.4.1	Aberration and Light-Time	119
6.4.2	Velocities of the Planets	121
6.4.3	Nutation	124
6.5	The PLANPOS Program	126
7	Physical Ephemerides of the Planets	131
7.1	Rotation	131
7.1.1	The Position Angle of the Axis	132
7.1.2	Planetographic Coordinates	135
7.2	Illumination Conditions	142
7.2.1	Phase and Elongation	142
7.2.2	The Position Angle of the Sun	144
7.2.3	Apparent Magnitude	144
7.2.4	Apparent Diameter	146
7.3	The PHYS Program	147
8	The Orbit of the Moon	153
8.1	General Description of the Lunar Orbit	153
8.2	Brown's Lunar Theory	157
8.3	The Chebyshev Approximation	167
8.4	The LUNA Program	173
9	Solar Eclipses	177
9.1	Times of New Moon	177
9.2	Geometry of an Eclipse	181
9.3	Geographic Coordinates and the Flattening of the Earth	185
9.4	Duration of an Eclipse	188
9.5	Solar and Lunar Coordinates	189
9.6	The ECLIPSE Program	191
9.7	Local Circumstances	199
9.8	The ECLTIMER Program	202

10 Stellar Occultations	205
10.1 Apparent Positions	206
10.2 Geocentric Conjunction	209
10.3 The Fundamental Plane	213
10.4 Disappearance and Reappearance	215
10.5 The OCCULT Program	217
10.6 Estimation of $\Delta T = ET - UT$ from Observations	226
11 Orbit Determination	229
11.1 Determining an Orbit from Two Position Vectors	229
11.1.1 The Sector-Triangle Ratio	230
11.1.2 Orbital Elements	233
11.2 The Geometry of Geocentric Observations	239
11.3 The Method of Successive Improvement	241
11.4 Multiple Solutions	243
11.5 The ORBDET Program	245
12 Astrometry	253
12.1 Photographic Imaging	253
12.2 Plate Constants	256
12.3 Reduction	257
12.4 The FOTO Program	260
Appendix	267
A.1 Notes on Alterations to Suit Individual Computers	267
A.2 List of Procedures	271
A.3 Procedures to Calculate Heliocentric Planetary Positions	273
Symbols	293
Glossary	297
Bibliography	301
Subject Index	307

1. Introduction

Recent years have seen a continuous increase in the power of small computers and a simultaneous decrease in their price. As a result many people interested in astronomy have such equipment at their disposal. This prompts the idea of using these computers for astronomical computation. What positive advantages are there in using one's own computer, when one can obtain all the most important data required for observing, fully as accurately, in one of the many yearly handbooks?

1.1 Some Examples

Let us first consider the calculation of the rising and setting times of the Sun and the Moon. Moving from one observation site to another differences in time soon occur, and these cannot be neglected. Yet in a handbook the rising and setting times are generally given for just a few places. If one wants the times at a given position, one is generally advised to use an interpolation routine or else read the corrections from nomograms. In this case it is significantly more convenient and more accurate if the computer can give the required data without further work.

The major portion of an almanac consists of pages and pages of tables of the positions of the Sun, the Moon, and the planets. Of all the information published, frequently only a small fraction is of specific interest. Using appropriate programs, just the values actually required can be calculated and simultaneously expressed in the desired coordinate system.

Even more important is the possibility of being able to calculate the orbit of a celestial body oneself. With new discoveries, actual ephemerides are frequently not available or only obtainable after some delay. A single set of orbital elements is sufficient for the motion of a comet to be followed with an adequate degree of accuracy.

Solar eclipses are impressive celestial phenomena, which often prompt amateur astronomers to undertake long journeys to observe them. Even though official sources may provide detailed information on a given eclipse, it is often desirable to plan a trip long before these data are published. If one requires information for a particular observing site that is not covered in the almanacs, a special program for computing local circumstances will be extremely useful.

Anyone who wants to observe stellar occultations by the Moon requires contact times for their observing site. When handbooks contain such predictions, these are generally restricted to a few selected major cities. Anyone wanting to observe from another site is once again reduced to resorting to approximations. The OCCULT program given in this book enables both the occurrence of stellar occultations and precise data about them to be calculated for any arbitrary selection of stars.

Many amateur astronomers are keen on astrophotography. Using reference stars with known coordinates, the position of a minor planet or a comet can be determined from photographs that one has taken oneself. If at least three exposures, taken at intervals, are available then the orbital elements may be calculated in the process of determining an orbit. In the programs PHOTO, ORBDET and COMET we provide the necessary tools for such highly complicated calculations that are required. Observers of minor planets will appreciate having the NUMINT program to use for computing precise ephemerides that take gravitational perturbations by the major planets into account, and that may be used even if the available orbital elements refer to an epoch that is several years ago.

1.2 Astronomy and Computing

Any introduction to the calculation of astronomical phenomena using computer programs would be incomplete without a discussion of the accuracy that can be attained. It is important, however, to first establish that the value of such programs primarily depends on the mathematical and physical description of the problem, as well as on the quality of the algorithms that are employed. It is often incorrectly assumed that the use of double-precision arithmetic will ensure the accuracy of a specific program. Yet even an accurately calculated program must return inaccurate results if the methods of calculation are not accurate in themselves.

An example of this is the calculation of ephemerides using the laws governing the two-body problem. Here the orbit of a planet is represented in a simplified manner by an ellipse. But apart from the gravitational force exerted by the Sun, which is responsible for this form of orbit, other forces are produced by the remaining planets. These lead to periodic and long-term perturbations, which may amount to about one arc-minute for the inner planets, and up to a degree for the outer planets. A theoretical Keplerian orbit is therefore only an approximation of the actual circumstances. If we require greater accuracy, then the use of more complicated descriptions of planetary orbits that model the mutual perturbations between the planets is unavoidable. Using high-precision arithmetic in the computer would not help in the slightest!

In developing our programs, we have tried to obtain an accuracy that is approximately the same as that found in astronomical yearbooks. The errors in the fundamental routines for determining the coordinates of the Sun, the Moon, and the planets amount to about $1''$ - $3''$. This accuracy is sufficient for the calculation of solar eclipses or stellar occultations, and should therefore be sufficient for most other purposes.

Anyone who wants to carry out accurate calculations consistently, however, must first be clear about the exact definitions of the coordinate systems used. Unfortunately, experience shows that values in different coordinate systems are cheerfully compared with one another. Then the cry goes up that the program is not accurate in its calculations, because different values are given in a yearbook. Frequently the discrepancy can be explained by the coordinates having a slightly different basis, which was not recognized by the user, who thus suspected an 'error'. We have therefore noted, at all relevant points, important corrections such as precession and nutation, or aberration and light-travel-time. The difference between Universal Time and Ephemeris Time, which repeatedly causes difficulties, is frequently discussed. Readers should not allow the frequent mention of these effects to confuse them. It is rather our intention that this form of discussion and its immediate application to appropriate programs will enable them to obtain a better feel of the magnitude and the practical significance of the individual corrections. The occasional repetitions appear to be necessary, because each chapter – and with it the description of a very specific topic – may also be read on its own. But as the programs are frequently constructed around the same routines, readers should also take the opportunity of reading other relevant chapters.

The power of a program not only depends on the mathematical description of the problem, but also on the use of appropriate computational methods. These are discussed at the time, in connection with the various applications. An example of this is the evaluation of angular functions in calculating the periodic perturbations in the orbit of the Moon and of the planets. By making use of the addition theorem and recursive relations for the trigonometric functions, the computing times required by the programs **PLANPOS** and **LUNA** are significantly reduced. If a particularly large number of positions, separated by short intervals, of a celestial body are required for some specific purpose, it is advisable to use Chebyshev polynomial approximations. It is then possible to obtain, without a loss of accuracy, valid expressions – covering a limited period of time – that can be particularly easily and quickly evaluated. Use of this technique is made in the programs **ECLIPSE**, **ECLTIMER** and **OCCULT**. Other aspects of mathematical computation are discussed in association with solving equations by Newton's procedure, *regula falsi*, and quadratic interpolation. In addition, a simple, but numerically stable algorithm for determining least-squares-fits is described in connection with the astrometric reduction of photographic plates.

1.3 Programming Languages and Techniques

In devising our programs, we endeavoured to follow certain basic rules, which can largely be said to be based on the concept of structured programming. We should perhaps explain in more detail what we understand by this.

First, the programs must be readable and understandable. Comments at important points in the program listing do not just serve to make it easier for the user to understand. Even the author of a program frequently forgets the details after a certain time, and the documentation then helps if the program has to be expanded at a later date. For the same reason, relevant variable names are advisable. Indentation of blocks of code or the use of blank lines makes it easier to appreciate the programs' logical structure. There is no reason to be sparing of blank space in any source code, because the compiler ignores it altogether.

As the concept of structured programming suggests, one should try to give a program the clearest possible logical structure. In practice each algorithm may be constructed from elementary structures that are statements, carry out sequences, make selections, or perform iterations (repetitions). Modern programming languages provide the necessary language elements to translate these structures into source code. This includes control structures (loops, conditional instructions), structured data types (records), and program blocks in the form of functions, sub-routines or separately compiled modules. Readers must decide for themselves how far unconditional jumps (GOTO-instructions) are necessary or useful.

If one wants to develop a program in this manner, then it is best to follow a systematic procedure. Modern software technology offers a variety of methods for a step-by-step structuring and sub-division of a given problem into smaller and smaller tasks, until these become easily comprehensible and may be simply transformed into code. The benefit of following a systematic approach is most pronounced for large and complex problems, because it helps to avoid redundancies and interface problems, assists in creating reusable solutions and saves a lot of development time. Direct coding without a detailed previous analysis of the problem is the most common error in software development. This method of working typically gives rise to programs that do not perform as expected and have to be modified by trial and error until they run successfully. Further hints on systematic software development techniques are given in the bibliography listed in the appendix.

If possible, a procedure designed for a specific task should be written in a universal way so that it can be used later in other programs. In producing astronomical programs, one repeatedly encounters certain basic problems, where this approach is particularly suitable. Examples of this are the evaluation of mathematical functions, the conversion of coordinates between different systems, or the determination of accurate positions for the planets, the Sun, or the Moon. It would be very laborious if one were forced to repeat

such processes every time. If, however, parts of programs that are repeatedly required are available in the form of a library of efficient, reliable sub-routines, then it is possible to concentrate on the functions required from the new portions of the programs, thus achieving the desired result much more easily and quickly.

The functions and procedures that we describe in this book form the basis for such a library of astronomical programs. The most important are

- technical and scientific functions, and auxiliary programs related to mathematical computation,
- routines for time and calendar calculations,
- various procedures concerning spherical astronomy,
- special routines for handling precession and nutation,
- procedures for calculating elliptical, parabolic, and hyperbolic Keplerian orbits, as well as
- routines for calculating accurate positions of the planets and the Moon, taking various perturbations into account.

A full list of the individual sub-routines is given in the Appendix.

In developing our programs we chose to use the Pascal language. An important reason was that Pascal is very widespread, and is implemented on all major systems from home computers to main-frames. Borland's Turbo-Pascal is a fast, inexpensive compiler that is available for all IBM-compatible personal computers. For computers in the Atari ST/TT series, which are quite popular in Germany, Application Systems Heidelberg offers the Turbo-Pascal-compatible Pure Pascal development kit. The source codes provided on the enclosed floppy disk may be used directly with both compilers. An adaptation to other dialects and standard Pascal should not, however, pose any difficulties, because the printed listings in the text do not contain Turbo-Pascal-specific language elements.

In contrast to Basic, which has numerous machine-specific dialects, Pascal programs can be written so that they are almost completely portable. If one tries to use only standard language elements in the programming, then the programs can be implemented on nearly any machine. Any changes necessary are insignificant, and can be made without problems. Some 'Notes on alterations to suit individual computers' appear in the Appendix. Full portability naturally involves some restrictions. For this reason, none of the programs provides graphic output, or uses any special features of the operating system. Any readers who may, for example, want to incorporate features of the computer's user interface (such as Windows), or have graphical output, can alter the programs to their own requirements whenever they wish.

Pascal was developed as a language for instruction and learning, and therefore demands a certain amount of programming discipline on the part

of users. Although people – and not only beginners – repeatedly complain about the ‘ponderous’ nature of the language, this does in fact conceal some distinct advantages. In particular, many unnecessary programming errors are avoided by one’s being forced to declare all variables specifically. In most versions of languages such as Fortran or Basic one can cheerfully be typing away, creating an unwanted variable with every typing error. Correcting these errors can be a severe test of one’s patience. A language like Pascal that requires declarations, discovers inconsistencies at compile time and warns the user with an appropriate error message. This is of particular importance in checking the various parameters required by a function or a sub-routine. Only when the types of variables agree with those that have been declared can the program be compiled. Forgetting an individual parameter is therefore impossible. This check by the compiler considerably simplifies the correct use of a library of previously created programs.

Unfortunately, this advantage of Pascal is accompanied by the disadvantage that a program must always be compiled as a whole. More highly developed languages such as Modula-2 or Ada handle the declaration of a sub-routine separately from the implementation. So they allow individual modules to be compiled separately, while retaining overall checking. This is of particular value in developing large groups of programs. Because of its similarities to Pascal, users of Modula-2, which has now become widely used on personal computers, will have little trouble adapting our programs.

Newer versions of Turbo-Pascal also offer similar possibilities called ‘units’. On the floppy disk accompanying this book we therefore provide some additional unit definition files for the user’s convenience. Aside from being necessary for running some of the larger programs with Turbo-Pascal, it facilitates the compilation and the use of our software as a standard library for astronomical calculations.

2. Coordinate Systems

Astronomy uses a whole series of different coordinate systems to specify the positions of stars and planets. Here we must distinguish between heliocentric coordinates, which are based on the Sun, and geocentric coordinates, which are centred on the Earth. Ecliptic coordinates refer the position of a point to the plane of the Earth's orbit, while equatorial coordinates are measured relative to the position of the Earth's or celestial equators. The slow shift of this reference plane because of precession means that the equinox of the coordinate system that is being used also has to be taken into account. Although it would obviously be possible to decide, once and for all, to use a single, fixed coordinate system, every system has specific advantages that make it particularly suitable for certain purposes.

Because there are various coordinate systems, it is often necessary to convert coordinates given in one system to those of another. The COCO ('Coordinate Conversion') program is designed to do just this. It provides accurate conversions between ecliptic and equatorial coordinates, and between geocentric and heliocentric coordinates, for various epochs. The individual transformations, which will be described shortly, are written as separate sub-routines. They can therefore be used elsewhere, and are an important basis for later programs.

2.1 Making a Start

Anyone who tries to write mathematical formulae in Pascal soon finds that some methods are unavailable, because—unlike Fortran or Basic—the language is very poorly provided with the required functions. This problem can be rapidly solved by including a few, suitable, short procedures and functions.

Trigonometric functions and their inverse functions are among those required, because they are either not available at all, or are only available in radians.

```
(*-----*)
(* SN: sine function (degrees) *)
(*-----*)
FUNCTION SN(X:REAL):REAL;
  CONST RAD=0.0174532925199433;
  BEGIN
```

```

    SN:=SIN(X*RAD)
END;

(*-----*)
(* CS: cosine function (degrees) *)
(*-----*)
FUNCTION CS(X:REAL):REAL;
    CONST RAD=0.0174532925199433;
    BEGIN
        CS:=COS(X*RAD)
    END;

(*-----*)
(* TN: tangent function (degrees) *)
(*-----*)
FUNCTION TN(X:REAL):REAL;
    CONST RAD=0.0174532925199433;
    VAR XX: REAL;
    BEGIN
        XX:=X*RAD; TN:=SIN(XX)/COS(XX);
    END;

(*-----*)
(* ASN: arcsine function (degrees) *)
(*-----*)
FUNCTION ASN(X:REAL):REAL;
    CONST RAD=0.0174532925199433; EPS=1E-7;
    BEGIN
        IF ABS(X)=1.0
            THEN ASN:=90.0*X
            ELSE IF (ABS(X)>EPS)
                THEN ASN:=ARCTAN(X/SQRT((1.0-X)*(1.0+X)))/RAD
                ELSE ASN:=X/RAD;
        END;

(*-----*)
(* ACS: arccosine function (degrees) *)
(*-----*)
FUNCTION ACS(X:REAL):REAL;
    CONST RAD=0.0174532925199433; EPS=1E-7; C=90.0;
    BEGIN
        IF ABS(X)=1.0
            THEN ACS:=C-X*C
            ELSE IF (ABS(X)>EPS)
                THEN ACS:=C-ARCTAN(X/SQRT((1.0-X)*(1.0+X)))/RAD
                ELSE ACS:=C-X/RAD;
        END;

(*-----*)
(* ATN: arctangent function (degrees) *)
(*-----*)
FUNCTION ATN(X:REAL):REAL;
    CONST RAD=0.0174532925199433;
    BEGIN
        ATN:=ARCTAN(X)/RAD
    END;
(*-----*)

```

The CUBR function calculates the cube root of its argument. It is used in later chapters and is only given here for the sake of completeness:

```
(*-----*)
(* CUBR: cube root *)
(*-----*)
FUNCTION CUBR(X:REAL):REAL;
  BEGIN
    IF (X=0.0) THEN CUBR:=0.0 ELSE CUBR:=EXP(LN(X)/3.0)
  END;
(*-----*)
```

A variant of the ATN function, which is also known in Fortran as ATN2, enables the arctangent of a fraction y/x to be calculated, but also simultaneously caters for the case where $x = 0$, when the fraction is undefined.

```
(*-----*)
(* ATN2: arctangent of y/x for two arguments *)
(*      (correct quadrant; -180 deg <= ATN2 <= +180 deg) *)
(*-----*)
FUNCTION ATN2(Y,X:REAL):REAL;
  CONST RAD=0.0174532925199433;
  VAR  AX,AY,PHI: REAL;
  BEGIN
    IF (X=0.0) AND (Y=0.0)
      THEN ATN2:=0.0
    ELSE
      BEGIN
        AX:=ABS(X); AY:=ABS(Y);
        IF (AX>AY)
          THEN PHI:=ARCTAN(AY/AX)/RAD
          ELSE PHI:=90.0-ARCTAN(AX/AY)/RAD;
        IF (X<0.0) THEN PHI:=180.0-PHI;
        IF (Y<0.0) THEN PHI:=-PHI;
        ATN2:=PHI;
      END;
    END;
  END;
(*-----*)
```

This function is used, in particular, for converting the plane Cartesian coordinates (x, y) of a point into the polar coordinates r and φ . Although $ATN(y/x)$ returns an angle between -90° and $+90^\circ$ (1^{st} and 4^{th} quadrants), $ATN2(y, x)$ takes into account the fact that for negative x , the angle φ corresponding to x and y lies between $+90^\circ$ and $+270^\circ$. Note that (as in Fortran) angles between 180° and 360° are expressed as negative values between -180° and 0° .

The conversion between the three-dimensional rectangular coordinates (x, y, z) and the polar coordinates (r, ϑ, φ) is slightly more complicated. The relationships between them (Fig. 2.1) are:

$$\begin{aligned} x &= r \cos \vartheta \cos \varphi & r &= \sqrt{x^2 + y^2 + z^2} \\ y &= r \cos \vartheta \sin \varphi & \tan \varphi &= y/x \\ z &= r \sin \vartheta & \tan \vartheta &= z / \sqrt{x^2 + y^2} \end{aligned} \quad (2.1)$$

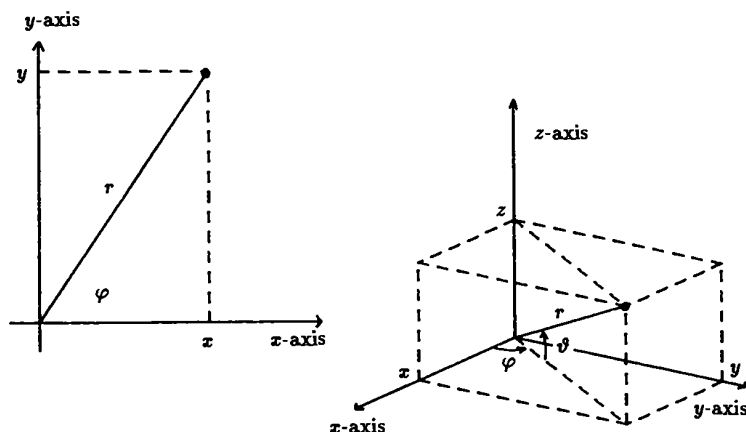


Fig. 2.1. Plane and spatial polar coordinates

The CART and POLAR procedures provide these frequently required conversions. The POLAR procedure is able to handle the exceptional cases where x and y are zero simultaneously, φ then being (arbitrarily) set equal to zero. According to the sign of z , ϑ is then equal to -90° , 0° or $+90^\circ$.

```
(*-----*)
(* CART: conversion of polar coordinates (r,theta,phi) *)
(* into cartesian coordinates (x,y,z) *)
(* (theta in [-90 deg,+90 deg]; phi in [-360 deg,+360 deg]) *)
(*-----*)
PROCEDURE CART(R,THETA,PHI: REAL; VAR X,Y,Z: REAL);
  VAR RCST : REAL;
  BEGIN
    RCST := R*CS(THETA);
    X := RCST*CS(PHI); Y := RCST*SN(PHI); Z := R*SN(THETA)
  END;
(*-----*)
(* POLAR: conversion of cartesian coordinates (x,y,z) *)
(* into polar coordinates (r,theta,phi) *)
(* (theta in [-90 deg,+90 deg]; phi in [0 deg,+360 deg]) *)
(*-----*)
PROCEDURE POLAR(X,Y,Z:REAL;VAR R,THETA,PHI:REAL);
  VAR RHO: REAL;
  BEGIN
    RHO:=X*X+Y*Y; R:=SQRT(RHO+Z*Z);
    PHI:=ATN2(Y,X); IF PHI<0 THEN PHI:=PHI+360.0;
    RHO:=SQRT(RHO); THETA:=ATN2(Z,RHO);
  END;
(*-----*)
```

In entering angles in conventional angular measurements, fractions of a degree are normally given in minutes and seconds of arc. This corresponds

to the usual method of expressing times in minutes and seconds. Conversion of these two forms is normally required no more than once in entering or returning a value. In principle, provided all values are positive this poses no problems. Nevertheless we need to take precautions to ensure that negative values are also correctly handled. The effect of the DDD and DMS procedures is best explained by a few examples:

DD	D M S
15.50000	15 30 00.0
-8.15278	-8 09 10.0
0.01667	0 1 0.0
-0.08334	0 -5 0.0

In converting a negative value DD into degrees, minutes and seconds, at any one time only the leading value of the three D, M and S is negative. It should be noted that in both procedures, D and M, which naturally can take only whole values, are defined as INTEGER while S is defined as REAL.

```
(*-----*)
(* DDD: conversion of degrees, minutes and seconds into *)
(* degrees and fractions of a degree *)
(*-----*)
PROCEDURE DDD(D,M:INTEGER;S:REAL;VAR DD:REAL);
  VAR SIGN: REAL;
  BEGIN
    IF ( (D<0) OR (M<0) OR (S<0) ) THEN SIGN:=-1.0 ELSE SIGN:=1.0;
    DD:=SIGN*(ABS(D)+ABS(M)/60.0+ABS(S)/3600.0);
  END;
(*-----*)
(* DMS: conversion of degrees and fractions of a degree *)
(* into degrees, minutes and seconds *)
(*-----*)
PROCEDURE DMS(DD:REAL;VAR D,M:INTEGER;VAR S:REAL);
  VAR D1:REAL;
  BEGIN
    D1:=ABS(DD); D:=TRUNC(D1);
    D1:=(D1-D)*60.0; M:=TRUNC(D1); S:=(D1-M)*60.0;
    IF (DD<0) THEN
      IF (D<>0) THEN D:=-D ELSE IF (M<>0) THEN M:=-M ELSE S:=-S;
    END;
  END;
(*-----*)
```

2.2 Calendar and Julian Dates

When calculating ephemerides, the difference in time between two given dates is generally required. A running day number, as has been commonly used in astronomy for a very long time, is therefore extremely convenient. The Julian Date gives the total number of days that have elapsed since 4713

B.C. January 1. It is named after Julius Scaliger, the father of Joseph Justus Scaliger, who was the first to use it for chronological purposes. As the count begins in biblical times, the Julian Date has now reached a very high value. At noon on 1980 July 23 it amounted to 2444 444.0 days, for example. When we recall that a second is approximately equal to 0.00001 day, then we find that we require a twelve-figure Julian Date to express an accurate time. The first two numbers hardly alter over a period of three centuries, however, and therefore the Modified Julian Date

$$\text{MJD} = \text{JD} - 2400000.5$$

is employed alongside the true Julian Date. The MJD is the total number of days that has elapsed since 1858 November 17 00:00 hours. Like the ordinary civil date, the MJD therefore changes at midnight, and not at noon like the Julian Date.

```
(*-----*)
(* MJD: Modified Julian Date *)
(* The routine is valid for any date since 4713 BC. *)
(* Julian calendar is used up to 1582 October 4, *)
(* Gregorian calendar is used from 1582 October 15 onwards. *)
(*-----*)
FUNCTION MJD(DAY,MONTH,YEAR:INTEGER;HOUR:REAL):REAL;
  VAR A: REAL; B: INTEGER;
  BEGIN
    A:=10000.0*YEAR+100.0*MONTH+DAY;
    IF (MONTH<=2) THEN BEGIN MONTH:=MONTH+12; YEAR:=YEAR-1 END;
    IF (A<=15821004.1)
      THEN B:=-2+TRUNC((YEAR+4716)/4)-1179
      ELSE B:=TRUNC(YEAR/400)-TRUNC(YEAR/100)+TRUNC(YEAR/4);
    A:=365.0*YEAR-679004.0;
    MJD:=A+B+TRUNC(30.6001*(MONTH+1))+DAY+HOUR/24.0;
  END;
(*-----*)
```

Here is an example:

```
VAR HOUR,MODJD,JD: REAL;
BEGIN
  DDD(3,30,10.0,HOUR); (* date 1961 Jan.14, 3:30:10 *)
  MODJD := MJD(14,1,1961,HOUR); (* Modified Julian Date *)
  JD := MODJD + 2400000.5; (* Julian Date *)
  WRITELN(' 1961 Jan.14, 3h30m10s:');
  WRITELN(' MJD = ', MODJD:20:5);
  WRITELN(' JD = ', JD:20:5);
END.
```

MJD takes into account the fact that, because of the Gregorian calendar reform, 1582 October 4 (JD 2299159.5) was followed by 1582 October 15 (JD 2299160.5). Until that date the Julian calendar is used, in which every fourth year included February 29. After that date, leap days are omitted in every

century year that is not evenly divisible by 400. As a result, the average length of the year according to the Gregorian calendar is

$$365 + 1/4 - 1/100 + 1/400 = 365.2425 \text{ days,}$$

slightly shorter than the 365.25 days of the Julian year.

CALDAT is a procedure to calculate the normal calendar date from the Modified Julian Date. Because it is closely linked to the subject of this section it will be given here.

```
(*-----*)
(* CALDAT: Finds the civil calendar date for a given value      *)
(*   of the Modified Julian Date (MJD).                        *)
(*   Julian calendar is used up to 1582 October 4,            *)
(*   Gregorian calendar is used from 1582 October 15 onwards. *)
(*-----*)
PROCEDURE CALDAT(MJD:REAL; VAR DAY,MONTH,YEAR:INTEGER;VAR HOUR:REAL);
  VAR B,D,F      : INTEGER;
      JD,JDO,C,E: REAL;
BEGIN
  JD := MJD + 2400000.5;
  (* JDO := TRUNC(JD+0.5);      *) (* Standard Pascal *)
  JDO := INT(JD+0.5);          (* TURBO Pascal   *)
  (* JDO := LONG_TRUNC(JD+0.5); *) (* ST Pascal plus *)
  IF (JDO<2299161.0)          (* calendar:      *)
    THEN BEGIN C:=JDO+1524.0 END (* -> Julian    *)
  ELSE BEGIN                  (* -> Gregorian *)
    B:=TRUNC((JDO-1867216.25)/36524.25);
    C:=JDO+(B-TRUNC(B/4))+1525.0
    END;
  D := TRUNC((C-122.1)/365.25);      E := 365.0*D+TRUNC(D/4);
  F := TRUNC((C-E)/30.6001);
  DAY := TRUNC(C-E+0.5)-TRUNC(30.6001*F); MONTH := F-1-12*TRUNC(F/14);
  YEAR := D-4715-TRUNC((7+MONTH)/10); HOUR := 24.0*(JD+0.5-JDO);
END;
(*-----*)
```

2.3 Ecliptic and Equatorial Coordinates

Ecliptic and equatorial coordinates differ in the reference plane from which they are measured. In the former case, the (x, y) plane is the Earth's orbital plane (the *ecliptic*), and in the latter, the plane perpendicular to the Earth's axis, i.e., parallel to the plane of the Earth's equator (Fig. 2.2). The x - x' -axis is common to both systems, and is defined as being the direction of the vernal equinox or *First Point of Aries*, designated by (Υ). This direction is perpendicular to the north pole of the ecliptic (the z -axis) and to the North Celestial Pole (the z' -axis). The angle ε between the ecliptic and the equator amounts to approximately $23^\circ 5'$.

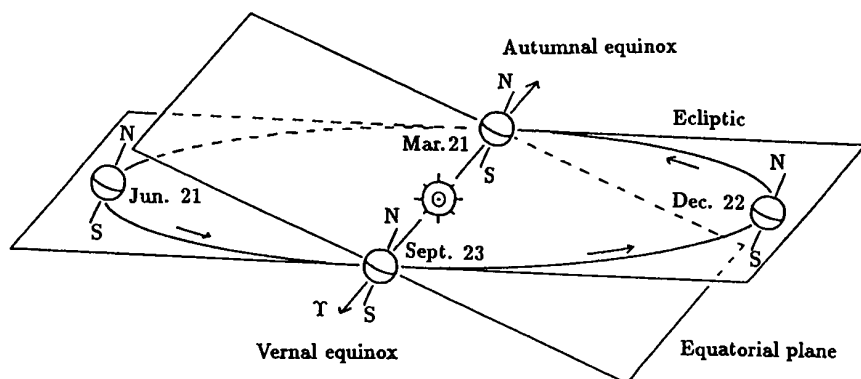


Fig. 2.2. Ecliptic and equator

What are the relationships between the ecliptic coordinates (x, y, z) and the equatorial coordinates (x', y', z') of a point? As $x = x'$, only the transformations $(y, z) \leftrightarrow (y', z')$ have to be considered. A point on the y -axis ($z = 0$) has equatorial coordinates $y' = +y \cos \epsilon$ and $z' = +y \sin \epsilon$. If the point lies on the z -axis, then its equatorial coordinates are $y' = -z \sin(\epsilon)$ and $z' = +z \cos \epsilon$. In the equatorial system, therefore, a given point (x, y, z) has the components

$$\begin{aligned} x' &= +x \\ y' &= +y \cos \epsilon - z \sin \epsilon \\ z' &= +y \sin \epsilon + z \cos \epsilon \end{aligned} \quad (2.2)$$

The corresponding inverse relationships are

$$\begin{aligned} x &= +x' \\ y &= +y' \cos \epsilon + z' \sin \epsilon \\ z &= -y' \sin \epsilon + z' \cos \epsilon \end{aligned} \quad (2.3)$$

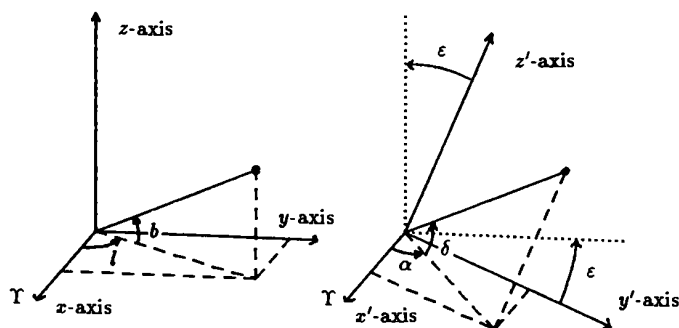


Fig. 2.3. Ecliptic and equatorial coordinates

The polar coordinates corresponding to (x, y, z) are ecliptic longitude l , ecliptic latitude b and distance r (see Fig. 2.3). In equatorial coordinates corresponding to these are right ascension α , declination δ , and again distance r , which has the same value in both systems.

$$\begin{aligned} x &= r \cos b \cos l & x' &= r \cos \delta \cos \alpha \\ y &= r \cos b \sin l & y' &= r \cos \delta \sin \alpha \\ z &= r \sin b & z' &= r \sin \delta \end{aligned} \quad (2.4)$$

From these two sets of equations it follows that the POLAR procedure may be used to derive longitude, latitude and distance, or right ascension and declination from Cartesian coordinates. Here is another small, example program:

```

VAR X,Y,Z,R,B,L,EPS,CEPS,SEPS,
    XPRIME,YPRIME,ZPRIME,ALPHA,DELTA: REAL;
BEGIN
  READ(X,Y,Z);
  POLAR(X,Y,Z,R,B,L);
  WRITELN(' Longitude: ',L,' Latitude: ',B,' Radius: ',R);
  EPS:=23.5; CEPS:=CS(EPS); SEPS:=SN(EPS);
  XPRIME := X;
  YPRIME := CEPS*Y-SEPS*Z;
  ZPRIME := SEPS*Y+CEPS*Z;
  POLAR(XPRIME,YPRIME,ZPRIME,R,DELTA,ALPHA);
  WRITELN(' Right Ascension: ',ALPHA/15.0,
    ' Declination: ',DELTA,' Radius: ',R);
END.
```

The value of right ascension will be given—as usual—in hours ($1^h \equiv 15^\circ$), instead of in degrees.

So far, nothing has been said about the exact value of the obliquity of the ecliptic ε . ε is not constant, but is decreasing by about $47''$ per century. This primarily arises from slow alterations in the Earth's orbit as a result of perturbations caused by the other planets. If we wish to avoid errors in coordinate transformations, it is particularly important whether the ecliptic for 1950 or for 2000 is used. An expression of the form 'equinox 2000' is therefore used, to specify that coordinates refer to the equinox and the ecliptic (or the equator) of the year 2000, for example. The exact value for the obliquity of the ecliptic as a function of time is

$$\varepsilon = 23^\circ 43' 29.111'' - 46''.8150 T - 0''.00059 T^2 + 0''.001813 T^3, \quad (2.5)$$

where T is the number of Julian centuries between the epoch and 2000 January 1 (12^h). In contrast to a century in the Gregorian calendar, which averages 36 524.25 days, a Julian century is always exactly 36 525 days. Important Julian epochs are J1900 (1900 January 0.5, JD 2415020.0) and J2000 (2000 January 1.5, JD 2451545.0). There is exactly one Julian century between

these dates. T may be calculated for a given epoch with the Julian Date JD from

$$T = \frac{JD - 2451545}{36525} .$$

To a close approximation, the value of T may be obtained from the year y by

$$T \approx (y - 2000)/100 .$$

```
(*-----*)
(* ECLEQU: Conversion of ecliptic into equatorial coordinates *)
(*      (T: equinox in Julian centuries since J2000) *)
(*-----*)
PROCEDURE ECLEQU(T:REAL;VAR X,Y,Z:REAL);
  VAR EPS,C,S,V: REAL;
  BEGIN
    EPS:=23.43929111-(46.8150+(0.00059-0.001813*T)*T)*T/3600.0;
    C:=CS(EPS); S:=SN(EPS);
    V:=+C*Y-S*Z; Z:=+S*Y+C*Z; Y:=V;
  END;
(*-----*)
(* EQUACL: conversion of equatorial into ecliptic coordinates *)
(*      (T: equinox in Julian centuries since J2000) *)
(*-----*)
PROCEDURE EQUACL(T:REAL;VAR X,Y,Z:REAL);
  VAR EPS,C,S,V: REAL;
  BEGIN
    EPS:=23.43929111-(46.8150+(0.00059-0.001813*T)*T)*T/3600.0;
    C:=CS(EPS); S:=SN(EPS);
    V:=+C*Y+S*Z; Z:=-S*Y+C*Z; Y:=V;
  END;
(*-----*)
```

The ECLEQU and EQUACL procedures are formulated so that the coordinates (x, y, z) entered are transformed into the appropriate system. If both routines are called one after the other (with the same value of T), then the original values are obtained.

The example given above may be expressed as follows using EQUACL (with Epoch J1950 = JD 2433282.50):

```
CONST JD2000 = 2451545.0;
      JD1950 = 2433282.5;
VAR X,Y,Z,R,B,L,
    T,DELTA,ALPHA: REAL;
BEGIN
  READ(X,Y,Z);
  POLAR(X,Y,Z,R,B,L);
  WRITELN(' Longitude: ',L,' Latitude: ',B,' Radius: ',R);
  T := (JD1950-JD2000) / 36525.0;
  ECLEQU(T,X,Y,Z);
```

```

POLAR(X,Y,Z,R,DELTA,ALPHA);
WRITELN(' Right Ascension: ',ALPHA/15.0,
        ' Declination: ',DELTA,' Radius: ', R);
END.

```

2.4 Precession

The shift in the position of the Earth's axis and of the ecliptic caused by forces exerted by the Sun, Moon and planets does not only cause a slight change in the angle ε between the equator and the ecliptic, but also a shift of the vernal equinox by about 1.5 per century (1' per year). For precise calculations, therefore, the equinox of the coordinate system used must be stated. The equinoxes most frequently used are

- the equinox of date,
- equinox J2000 and
- equinox B1950.

"Equinox of date" means that the values used are those for the equator, ecliptic, and vernal equinox for the actual date under consideration. Such daily alteration of the coordinate system is sensible if one requires the coordinates of a planet, for example, for use in conjunction with the setting circles on an equatorially mounted telescope, or on a transit circle. Because of the shift in the Earth's axis, the orientation of the polar axis of a telescope will also alter. On the other hand, if one wants to study the actual spatial motion of a planet, then it is better to use a fixed equinox, such as that for Julian Epoch J2000 (2000 January 1.5 = JD 2451545.0), which was generally introduced in 1984. Before that, the older equinox B1950 had been used for a long time, and was employed for many stellar catalogues and atlases (such as the *SAO Star Catalog* and *Atlas Coeli*). The prefix B indicates that it is not half-way between the epochs J1900 and J2000 (which would, of course, be J1950 = JD 2433282.5), but the beginning of the Besselian year 1950 (1950 Jan. 0.923 = JD 2433282.423).

Transformation of coordinates from one equinox to another requires similar steps to those required in converting from ecliptic to equatorial coordinates. Whereas those two coordinates systems could be brought into coincidence by a simple rotation about the x -axis, we now have to deal with a total of three rotations: first about the z -axis, second about the x -axis, and finally about the z -axis again.

If we first consider the ecliptic at time T_0 and time $T_0 + T$, then the angle between the two planes is

$$\pi = (47''0029 - 0''06603 \cdot T_0 + 0''000598 \cdot T_0^2) \cdot T \quad (2.6)$$

$$+ (-0''03302 + 0''000598 \cdot T_0) \cdot T^2 + 0''000060 \cdot T^3$$

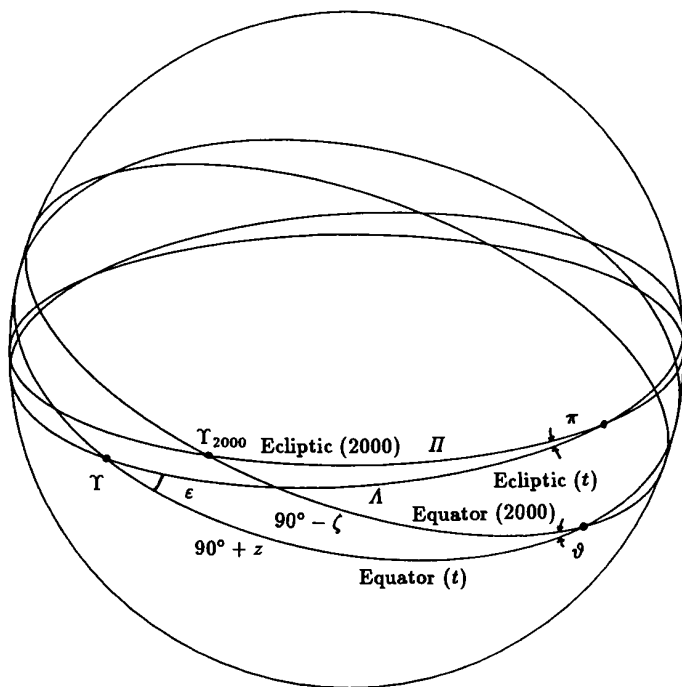


Fig. 2.4. The effects of precession on the ecliptic, equator, and vernal equinox

If two coordinate systems (x', y', z') and (x'', y'', z'') are established, such that the x' - y' -plane coincides with the ecliptic for time T_0 , and the x'' - y'' -plane with that for time $T_0 + T$, then we have:

$$\begin{aligned} x'' &= x' \\ y'' &= +\cos \pi \cdot y' + \sin \pi \cdot z' \\ z'' &= -\sin \pi \cdot y' + \cos \pi \cdot z' \end{aligned}$$

This assumes that the x' -axis and the x'' -axis coincide with the line of intersection of the two planes. Let (x_0, y_0, z_0) be the ecliptic coordinates corresponding to the epoch T_0 and (x, y, z) the coordinates at epoch $T_0 + T$. We then have

$$\begin{aligned} x' &= +\cos \Pi \cdot x_0 + \sin \Pi \cdot y_0 \\ y' &= -\sin \Pi \cdot x_0 + \cos \Pi \cdot y_0 \\ z' &= z_0 \end{aligned}$$

and

$$x = +\cos \Lambda \cdot x'' - \sin \Lambda \cdot y''$$

$$\begin{aligned} y &= +\sin \Lambda \cdot x'' + \cos \Lambda \cdot y'' \\ z &= z'' \quad , \end{aligned}$$

where

$$\begin{aligned} \Pi &= (174^{\circ}876383889 + 3289''.4789T_0 + 0''.60622T_0^2) \\ &\quad + (-869''.8089 - 0''.50491T_0)T + 0''.03536T^2 \\ p &= (5029''.0966 + 2''.22226T_0 - 0''.000042T_0^2)T \\ &\quad + (1''.11113 - 0''.000042T_0)T^2 - 0''.000006T^3 \\ \Lambda &= \Pi + p \quad . \end{aligned} \quad (2.7)$$

Π and Λ are the angles between the x'/x'' -axis and the vernal equinox Υ_0 at epoch T_0 (x_0 -axis), and the vernal equinox Υ at epoch $T_0 + T$ (x -axis), respectively. p denotes precession in longitude, because the change in the vernal equinox mainly takes place in ecliptic longitude. Substituting for the various relationships, we obtain the equations

$$\begin{aligned} x &= a_{11} \cdot x_0 + a_{12} \cdot y_0 + a_{13} \cdot z_0 \\ y &= a_{21} \cdot x_0 + a_{22} \cdot y_0 + a_{23} \cdot z_0 \\ z &= a_{31} \cdot x_0 + a_{32} \cdot y_0 + a_{33} \cdot z_0 \end{aligned} \quad (2.8)$$

where

$$\begin{aligned} a_{11} &= +\cos \Lambda \cos \Pi + \sin \Lambda \cos \pi \sin \Pi \\ a_{21} &= +\sin \Lambda \cos \Pi - \cos \Lambda \cos \pi \sin \Pi \\ a_{31} &= +\sin \pi \sin \Pi \\ a_{12} &= +\cos \Lambda \sin \Pi - \sin \Lambda \cos \pi \cos \Pi \\ a_{22} &= +\sin \Lambda \sin \Pi + \cos \Lambda \cos \pi \cos \Pi \\ a_{32} &= -\sin \pi \cos \Pi \\ a_{13} &= -\sin \Lambda \sin \pi \\ a_{23} &= +\cos \Lambda \sin \pi \\ a_{33} &= +\cos \pi \quad . \end{aligned} \quad (2.9)$$

The angles in equatorial coordinates corresponding to the three angles π , Π and Λ are $90^\circ - \zeta$, ϑ and $90^\circ + z$:

$$\begin{aligned} \zeta &= (2306''.2181 + 1''.39656T_0 - 0''.000139T_0^2)T \\ &\quad + (0''.30188 - 0''.000345T_0)T^2 + 0''.017998T^3 \\ \vartheta &= (2004''.3109 - 0''.85330T_0 - 0''.000217T_0^2)T \\ &\quad + (-0''.42665 - 0''.000217T_0)T^2 - 0''.041833T^3 \\ z &= \zeta + (0''.79280 + 0''.000411T_0)T^2 + 0''.000205T^3 \quad . \end{aligned} \quad (2.10)$$

This gives corresponding formulae where

$$\begin{aligned}
 a_{11} &= -\sin z \sin \zeta + \cos z \cos \vartheta \cos \zeta \\
 a_{21} &= +\cos z \sin \zeta + \sin z \cos \vartheta \cos \zeta \\
 a_{31} &= +\sin \vartheta \cos \zeta \\
 a_{12} &= -\sin z \cos \zeta - \cos z \cos \vartheta \sin \zeta \\
 a_{22} &= +\cos z \cos \zeta - \sin z \cos \vartheta \sin \zeta \\
 a_{32} &= -\sin \vartheta \sin \zeta \\
 a_{13} &= -\cos z \sin \vartheta \\
 a_{23} &= -\sin z \sin \vartheta \\
 a_{33} &= +\cos \vartheta
 \end{aligned} \tag{2.11}$$

The most onerous part of calculating precession is determining the values for the various angles and for a_{ij} . However, these values only depend on the two epochs T_0 and $T_0 + T$, and can be reused if a whole series of different positions are to be converted from one fixed epoch to another. Individual procedures are therefore given for this purpose.

```

(*-----*)
(* PMATECL: calculates the precession matrix A[i,j] for          *)
(*      transforming ecliptic coordinates from equinox T1 to T2 *)
(*      ( T=(JD-2451545.0)/36525 )                               *)
(*-----*)
PROCEDURE PMATECL(T1,T2:REAL;VAR A: REAL33);
  CONST SEC=3600.0;
  VAR DT,PPI,PI,PA: REAL;
  C1,S1,C2,S2,C3,S3: REAL;
  BEGIN
    DT:=T2-T1;
    PPI := 174.876383889 + ( ((3289.4789+0.60622*T1)*T1) +
      ((-869.8089-0.50491*T1) + 0.03536*DT)*DT )/SEC;
    PI  := ( (47.0029-(0.06603-0.000598*T1)*T1)+
      ((-0.03302+0.000598*T1)+0.000060*DT)*DT )/SEC;
    PA  := ( (5029.0966+(2.22226-0.000042*T1)*T1)+
      ((1.11113-0.000042*T1)-0.000006*DT)*DT )/SEC;
    C1:=CS(PPI+PA); C2:=CS(PI); C3:=CS(PPI);
    S1:=SN(PPI+PA); S2:=SN(PI); S3:=SN(PPI);
    A[1,1]:=+C1*C3+S1*C2*S3; A[1,2]:=+C1*S3-S1*C2*C3; A[1,3]:=-S1*S2;
    A[2,1]:=+S1*C3-C1*C2*S3; A[2,2]:=+S1*S3+C1*C2*C3; A[2,3]:=+C1*S2;
    A[3,1]:=+S2*S3; A[3,2]:=-S2*C3; A[3,3]:=+C2;
  END;

(*-----*)
(* PMATEQU: calculates the precession matrix A[i,j] for          *)
(*      transforming equatorial coordinates from equinox T1 to T2 *)
(*      (T=(JD-2451545.0)/36525 )                               *)
(*-----*)
PROCEDURE PMATEQU(T1,T2:REAL;VAR A:REAL33);

```

```

CONST SEC=3600.0;
VAR DT,ZETA,Z,THETA: REAL;
      C1,S1,C2,S2,C3,S3: REAL;
BEGIN
  DT:=T2-T1;
  ZETA := ( (2306.2181+(1.39656-0.000139*T1)*T1)+
    ((0.30188-0.000345*T1)+0.017998*DT)*DT )/DT/SEC;
  Z      := ZETA + ( (0.79280+0.000411*T1)+0.000205*DT)*DT/DT/SEC;
  THETA := ( (2004.3109-(0.85330+0.000217*T1)*T1)-
    ((0.42665+0.000217*T1)+0.041833*DT)*DT )/DT/SEC;
  C1:=CS(Z); C2:=CS(THETA); C3:=CS(ZETA);
  S1:=SN(Z); S2:=SN(THETA); S3:=SN(ZETA);
  A[1,1]:=-S1*S3+C1*C2*C3; A[1,2]:=-S1*C3-C1*C2*S3; A[1,3]:=-C1*S2;
  A[2,1]:=-C1*S3+S1*C2*C3; A[2,2]:=-C1*C3-S1*C2*S3; A[2,3]:=-S1*S2;
  A[3,1]:=-S2*C3; A[3,2]:=-S2*S3; A[3,3]:=-C2;
END;

(*-----*)
(* PRECART: calculate change of coordinates due to precession *)
(* for given transformation matrix A[i,j] *)
(* (to be used with PMATECL und PMATEQU) *)
(*-----*)
PROCEDURE PRECART(A:REAL33; VAR X,Y,Z:REAL);
  VAR U,V,W: REAL;
  BEGIN
    U := A[1,1]*X+A[1,2]*Y+A[1,3]*Z;
    V := A[2,1]*X+A[2,2]*Y+A[2,3]*Z;
    W := A[3,1]*X+A[3,2]*Y+A[3,3]*Z;
    X:=U; Y:=V; Z:=W;
  END;
(*-----*)

```

The matrix A is defined as data type REAL33. A corresponding global type declaration

```
TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
```

should be included in the initialization section of the main program.

In order to give an idea of the effects of precession, and also to allow some practice in using the various routines, a somewhat longer example will be given. A set of coordinates for equinox B1950 is to be converted to a corresponding set for equinox J2000. This may be carried out in two ways, according to whether precession or the conversion between ecliptic and equatorial coordinates is applied first.

```

CONST B1950 = -0.500002108; (* B1950=(2433282.423-2451545)/36525 *)
      J2000 = 0.0;

VAR   A           : REAL33;
      R,B,L,X0,Y0,Z0,
      X,Y,Z,DEC,RA : REAL;
      I           : INTEGER;

```

BEGIN

```
(* ecliptic coordinates B1950 (angles in degrees): *)
L := 200.0; B:=10.0; R:=1.0;
CART (R,B,L,X0,Y0,Z0);

(* method 1: *)
PMATECL(B1950,J2000,A); (* Matrix B1950->J2000 ecliptic: *)
FOR I:=1 TO 3 DO WRITELN(A[I,1]:15:10,A[I,2]:15:10,A[I,3]:15:10);
X:=X0; Y:=Y0; Z:=Z0;      (* ecl. B1950 *)
PRECART(A,X,Y,Z);          (* ecl. J2000 *)
POLAR(X,Y,Z,R,B,L);
WRITELN(L:15:10,B:15:10,R:15:10);
ECLEQU(J2000,X,Y,Z);      (* equ. J2000 *)
POLAR(X,Y,Z,R,DEC,RA);
WRITELN(RA/15.0:15:10,DEC:15:10,R:15:10);

(* method 2: *)
PMATEQU(B1950,J2000,A); (* Matrix B1950->J2000 equatorial: *)
FOR I:=1 TO 3 DO WRITELN(A[I,1]:15:10,A[I,2]:15:10,A[I,3]:15:10);
X:=X0; Y:=Y0; Z:=Z0;      (* ecl. B1950 *)
ECLEQU(B1950,X,Y,Z);      (* equ. B1950 *)
POLAR(X,Y,Z,R,DEC,RA);
WRITELN(RA/15.0:15:10,DEC:15:10,R:15:10);
PRECART(A,X,Y,Z);          (* equ. J2000 *)
POLAR(X,Y,Z,R,DEC,RA);
WRITELN(RA/15.0:15:10,DEC:15:10,R:15:10);
```

END

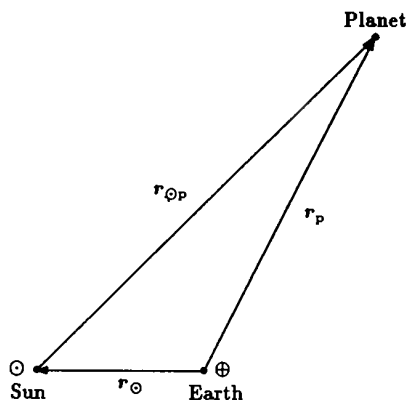


Fig. 2.5. The Earth-Sun-Planet triangle

2.5 Geocentric Coordinates and the Orbit of the Sun

All that is now required to complete the program is a method for converting heliocentric coordinates (referred to the centre of the Sun) into geocentric ones (referred to the centre of the Earth), and the converse. Conversion of the origin of the coordinates is described by the equations

$$\mathbf{r}_p = \mathbf{r}_{\odot p} + \mathbf{r}_{\odot} \quad \text{and} \quad \mathbf{r}_{\odot p} = \mathbf{r}_p - \mathbf{r}_{\odot} \quad (2.12)$$

which are derived from the Sun-Earth-Planet triangle illustrated in Fig. 2.5. Here $\mathbf{r}_{\odot p}$ and $\mathbf{r}_p$ are the heliocentric and geocentric position vectors of a point P , and $\mathbf{r}_{\odot}$ is the geocentric position of the Sun. Written in the form of individual components, the transformation equations are

$$\begin{aligned} x_p &= x_{\odot p} + x_{\odot} & x_{\odot p} &= x_p - x_{\odot} \\ y_p &= y_{\odot p} + y_{\odot} & y_{\odot p} &= y_p - y_{\odot} \\ z_p &= z_{\odot p} + z_{\odot} & z_{\odot p} &= z_p - z_{\odot} \end{aligned} ,$$

where it makes no difference whether ecliptic or equatorial coordinates are employed to express the various vectors.

As any error in the position of the Sun inevitably affects the accuracy of the conversions, it is worthwhile taking some trouble in calculating the coordinates of the Sun. The SUN200 procedure, introduced here, attains an accuracy of about $1''$, which should be more than sufficient for most purposes. For a time

$$T = (\text{JD} - 2451545)/36525$$

it provides the ecliptic longitude (L) and latitude (B) of the Sun, and its distance (R) from the Earth. From these we can determine the ecliptic coordinates

$$\begin{aligned} x_{\odot} &= R \cos B \cos L \\ y_{\odot} &= R \cos B \sin L \\ z_{\odot} &= R \sin B \end{aligned} .$$

Like L and B these initially refer to the equinox of date, but can be converted to another equinox by means of the procedures already discussed.

```
(*-----*)
(* SUN200: ecliptic coordinates L,B,R (in deg and AU) of the *)
(*      Sun referred to the mean equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      (   = (JED-2451545.0)/36525 ) *)
(*-----*)
PROCEDURE SUN200(T:REAL;VAR L,B,R:REAL);
```

```
CONST P2=6.283185307;
```

```

VAR C3,S3:          ARRAY [-1..7] OF REAL;
    C,S:            ARRAY [-8..0] OF REAL;
    M2,M3,M4,M5,M6: REAL;
    D,A,UU:         REAL;
    U,V,DL,DR,DB:   REAL;
    I:              INTEGER;

FUNCTION FRAC(X:REAL):REAL;
    (* for some compilers TRUNC has to be replaced by LONG_TRUNC *)
    (* or INT (Turbo-Pascal) if the routine is used with T<-24 *)
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1+C2-S1*S2; S:=S1+C2+C1*S2; END;

PROCEDURE TERM(I1,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
        IF IT=0 THEN ADDTHE(C3[I1],S3[I1],C[I],S[I],U,V)
        ELSE BEGIN U:=U*T; V:=V*T END;
        DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

PROCEDURE PERTVEN; (* Keplerian terms and perturbations by Venus *)
    VAR I: INTEGER;
    BEGIN
        C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M2); S[-1]:=-SIN(M2);
        FOR I:=-1 DOWNT0 -5 DO ADDTHE(C[I],S[I],C[-I],S[-I],C[I-1],S[I-1]);
        TERM(1, 0,0,-0.22,6892.76,-16707.37, -0.54, 0.00, 0.00);
        TERM(1, 0,1,-0.06, -17.35, 42.04, -0.15, 0.00, 0.00);
        TERM(1, 0,2,-0.01, -0.05, 0.13, -0.02, 0.00, 0.00);
        TERM(2, 0,0, 0.00, 71.98, -139.57, 0.00, 0.00, 0.00);
        TERM(2, 0,1, 0.00, -0.36, 0.70, 0.00, 0.00, 0.00);
        TERM(3, 0,0, 0.00, 1.04, -1.75, 0.00, 0.00, 0.00);
        TERM(0,-1,0, 0.03, -0.07, -0.16, -0.07, 0.02,-0.02);
        TERM(1,-1,0, 2.35, -4.23, -4.75, -2.64, 0.00, 0.00);
        TERM(1,-2,0,-0.10, 0.06, 0.12, 0.20, 0.02, 0.00);
        TERM(2,-1,0,-0.06, -0.03, 0.20, -0.01, 0.01,-0.09);
        TERM(2,-2,0,-4.70, 2.90, 8.28, 13.42, 0.01,-0.01);
        TERM(3,-2,0, 1.80, -1.74, -1.44, -1.57, 0.04,-0.06);
        TERM(3,-3,0,-0.67, 0.03, 0.11, 2.43, 0.01, 0.00);
        TERM(4,-2,0, 0.03, -0.03, 0.10, 0.09, 0.01,-0.01);
        TERM(4,-3,0, 1.51, -0.40, -0.88, -3.36, 0.18,-0.10);
        TERM(4,-4,0,-0.19, -0.09, -0.38, 0.77, 0.00, 0.00);
        TERM(5,-3,0, 0.76, -0.68, 0.30, 0.37, 0.01, 0.00);
        TERM(5,-4,0,-0.14, -0.04, -0.11, 0.43,-0.03, 0.00);
        TERM(5,-5,0,-0.05, -0.07, -0.31, 0.21, 0.00, 0.00);
        TERM(6,-4,0, 0.15, -0.04, -0.06, -0.21, 0.01, 0.00);
        TERM(6,-5,0,-0.03, -0.03, -0.09, 0.09,-0.01, 0.00);
        TERM(6,-6,0, 0.00, -0.04, -0.18, 0.02, 0.00, 0.00);
        TERM(7,-5,0,-0.12, -0.03, -0.08, 0.31,-0.02,-0.01);
    END;

PROCEDURE PERTMAR; (* perturbations by Mars *)

```

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M4); S[-1]:=-SIN(M4);

FOR I:=-1 DOWNT0 -7 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(1,-1,0,-0.22, 0.17, -0.21, -0.27, 0.00, 0.00);

TERM(1,-2,0,-1.66, 0.62, 0.16, 0.28, 0.00, 0.00);

TERM(2,-2,0, 1.96, 0.57, -1.32, 4.55, 0.00, 0.01);

TERM(2,-3,0, 0.40, 0.15, -0.17, 0.46, 0.00, 0.00);

TERM(2,-4,0, 0.53, 0.26, 0.09, -0.22, 0.00, 0.00);

TERM(3,-3,0, 0.05, 0.12, -0.35, 0.15, 0.00, 0.00);

TERM(3,-4,0,-0.13, -0.48, 1.06, -0.29, 0.01, 0.00);

TERM(3,-5,0,-0.04, -0.20, 0.20, -0.04, 0.00, 0.00);

TERM(4,-4,0, 0.00, -0.03, 0.10, 0.04, 0.00, 0.00);

TERM(4,-5,0, 0.05, -0.07, 0.20, 0.14, 0.00, 0.00);

TERM(4,-6,0,-0.10, 0.11, -0.23, -0.22, 0.00, 0.00);

TERM(5,-7,0,-0.05, 0.00, 0.01, -0.14, 0.00, 0.00);

TERM(5,-8,0, 0.05, 0.01, -0.02, 0.10, 0.00, 0.00);

END;

PROCEDURE PERTJUP; (* perturbations by Jupiter *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M5); S[-1]:=-SIN(M5);

FOR I:=-1 DOWNT0 -3 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(-1,-1,0,0.01, 0.07, 0.18, -0.02, 0.00,-0.02);

TERM(0,-1,0,-0.31, 2.58, 0.52, 0.34, 0.02, 0.00);

TERM(1,-1,0,-7.21, -0.06, 0.13,-16.27, 0.00,-0.02);

TERM(1,-2,0,-0.54, -1.52, 3.09, -1.12, 0.01,-0.17);

TERM(1,-3,0,-0.03, -0.21, 0.38, -0.06, 0.00,-0.02);

TERM(2,-1,0,-0.16, 0.05, -0.18, -0.31, 0.01, 0.00);

TERM(2,-2,0, 0.14, -2.73, 9.23, 0.48, 0.00, 0.00);

TERM(2,-3,0, 0.07, -0.55, 1.83, 0.25, 0.01, 0.00);

TERM(2,-4,0, 0.02, -0.08, 0.25, 0.06, 0.00, 0.00);

TERM(3,-2,0, 0.01, -0.07, 0.16, 0.04, 0.00, 0.00);

TERM(3,-3,0,-0.16, -0.03, 0.08, -0.64, 0.00, 0.00);

TERM(3,-4,0,-0.04, -0.01, 0.03, -0.17, 0.00, 0.00);

END;

PROCEDURE PERTSAT; (* perturbations by Saturn *)

BEGIN

C[-1]:=COS(M6); S[-1]:=-SIN(M6);

ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);

TERM(0,-1,0, 0.00, 0.32, 0.01, 0.00, 0.00, 0.00);

TERM(1,-1,0,-0.08, -0.41, 0.97, -0.18, 0.00,-0.01);

TERM(1,-2,0, 0.04, 0.10, -0.23, 0.10, 0.00, 0.00);

TERM(2,-2,0, 0.04, 0.10, -0.35, 0.13, 0.00, 0.00);

END;

PROCEDURE PERTMOO; (* difference between the Earth-Moon *)

BEGIN

(* barycenter and the center of the Earth *)

DL := DL + 6.45*SIN(D) - 0.42*SIN(D-A) + 0.18*SIN(D+A)
+ 0.17*SIN(D-M3) - 0.06*SIN(D+M3);

DR := DR + 30.76*COS(D) - 3.06*COS(D-A) + 0.85*COS(D+A)
- 0.58*COS(D+M3) + 0.57*COS(D-M3);

```

      DB := DB + 0.576*SIN(UU);
    END;

  BEGIN (* SUN200 *)
    DL:=0.0; DR:=0.0; DB:=0.0;
    M2:=P2*FRAC(0.1387306+162.5485917*T);
    M3:=P2*FRAC(0.9931266+99.9973604*T);
    M4:=P2*FRAC(0.0543250+ 53.1666028*T);
    M5:=P2*FRAC(0.0551750+ 8.4293972*T);
    M6:=P2*FRAC(0.8816500+ 3.3938722*T); D :=P2*FRAC(0.8274+1236.8531*T);
    A :=P2*FRAC(0.3749+1325.5524*T);    UU:=P2*FRAC(0.2591+1342.2278*T);
    C3[0]:=1.0;    S3[0]:=0.0;
    C3[1]:=COS(M3); S3[1]:=SIN(M3); C3[-1]:=C3[1]; S3[-1]:=-S3[1];
    FOR I:=2 TO 7 DO ADDTHE(C3[I-1],S3[I-1],C3[1],S3[1],C3[I],S3[I]);
    PERTVEN; PERTMAR; PERTJUP; PERTSAT; PERTMOO;
    DL:=DL + 6.40*SIN(P2*(0.6983+0.0561*T))+1.87*SIN(P2*(0.5764+0.4174*T))
      + 0.27*SIN(P2*(0.4189+0.3306*T))+0.20*SIN(P2*(0.3581+2.4814*T));
    L:= 360.0*FRAC(0.7859453 + M3/P2 + ((6191.2+1.1*T)*T+DL)/1296.0E3 );
    R:= 1.0001398 - 0.0000007*T + DR*1E-6;
    B:= DB/3600.0;
  END;  (* SUN200 *)
  (*-----*)

```

The extent of the SUN200 procedure is explained by the fact that the relative motion of the Sun and the Earth cannot be described to the required accuracy by simple elliptical orbits. Apart from the perturbations of the other planets—particularly those of Venus and Jupiter—there is also the monthly oscillation of the centre of the Earth around the barycentre of the Earth-Moon system. Strictly speaking, not the Earth itself, but rather the position of the barycentre, is moving in the plane of the ecliptic. The Moon's orbit around the Earth is therefore reflected in a small periodic perturbation of the geocentric orbit of the Sun. The details of the procedure and its basis will not be discussed here. Both will be treated thoroughly in Chap. 6, where appropriate routines for all the planets will be given.

2.6 The COCO Program

The various procedures described will now be incorporated into a complete program. Only the input and output routines and the main program are given here. The procedures already discussed should be incorporated at the points indicated.

```
(*-----*)
(*                                COCO                                *)
(*                                coordinate conversion                  *)
(*                                version 93/07/01                      *)
(*-----*)
```

```
PROGRAM COCO(INPUT,OUTPUT);
```

```
    TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
```

```
    VAR X,Y,Z,XS,YS,ZS: REAL;
        T,TEQX,TEQIN  : REAL;
        LS,BS,RS      : REAL;
        A              : REAL33;
        ECLIPT         : BOOLEAN;
        MODE           : CHAR;
```

```
(*-----*)
(* The following procedures should be entered here in the given order: *)
(* SN, CS, ATN, ATN2, CART, POLAR, DDD, DMS                             *)
(* MJD                                                                    *)
(* ECLEQU, EQUECL                                                         *)
(* PMATECL, PMATEQU, PRECART, SUN200                                     *)
(*-----*)
```

```
(*-----*)
PROCEDURE GETEQX(VAR TEQX:REAL);
    BEGIN
        WRITE ( ' equinox (yyyy) ? ');
        READLN (TEQX); TEQX := (TEQX-2000.0)/100.0;
    END;
```

```
(*-----*)
PROCEDURE GETDAT(VAR T:REAL);
    VAR O,M,Y : INTEGER;
        HOUR,JD: REAL;
    BEGIN
        WRITE ( ' Date (year month day hour) ? ');
        READLN (Y,M,D,HOUR);
        JD := MJD(D,M,Y,HOUR) + 2400000.5;
        WRITELN; WRITELN ( ' JD',JD:13:4); WRITELN;
        T := (JD-2451545.0) / 36525.0;
    END;
```

(*-----*)

PROCEDURE GETINP (VAR X,Y,Z,TEQX:REAL;VAR ECLIPT:BOOLEAN);

VAR I,D,M : INTEGER;
L,B,R,S: REAL;

BEGIN (* GETINP *)

```

WRITELN;
WRITELN('          COCO: coordinate conversion          ');
WRITELN('          version 93/07/01                          ');
WRITELN('          (c) 1993 Thomas Pfleger, Oliver Montenbruck ');
WRITELN;
WRITELN(' Coordinate input: please select format required ');
WRITELN;
WRITELN(' 1  ecliptic  cartesian      2  ecliptic  polar');
WRITELN(' 3  equatorial cartesian    4  equatorial polar');
WRITELN;
WRITE (' '); READLN (I); WRITELN;

```

CASE I OF

```

1: BEGIN
    WRITE (' Coordinates (x y z) ? '); READLN(X,Y,Z); ECLIPT:=TRUE;
    END;
2: BEGIN
    WRITE (' Coordinates (L (o '' ") B (o '' ") R) ? ');
    READ(D,M,S); DDD(D,M,S,L); READLN(D,M,S,R); DDD(D,M,S,B);
    CART(R,B,L,X,Y,Z); ECLIPT:=TRUE;
    END;
3: BEGIN
    WRITE (' Coordinates (x y z) ? '); READLN(X,Y,Z);ECLIPT:=FALSE;
    END;
4: BEGIN
    WRITE (' Coordinates (RA (h m s) DEC (o '' ") R) ? ');
    READ(D,M,S); DDD(D,M,S,L); READLN(D,M,S,R); DDD(D,M,S,B);
    L:=L*15.0; CART(R,B,L,X,Y,Z); ECLIPT:=FALSE;
    END;
END; (* CASE *)

```

GETEQX (TEQX); (* read equinox *)

END; (* GETINP *)

(*-----*)

PROCEDURE RESULT (X,Y,Z: REAL; ECLIPT: BOOLEAN);

VAR L,B,R,S: REAL;
D,M : INTEGER;

BEGIN (* RESULT *)

```
WRITELN; WRITELN (' (x,y,z) = (' ,x:13:8,',',y:13:8,',',z:13:8,')');
WRITELN;
```

```
POLAR (X,Y,Z,R,B,L);
```

```
IF ECLIPT
```

```
THEN
```

```
BEGIN
```

```
WRITELN (' ' :5,' o ' ' ' ' :8,' o ' ' ' ');
```

```
DMS(L,D,M,S); WRITE(' L = ',D:3,M:3,S:5:1,' ' :3);
```

```
DMS(B,D,M,S); WRITE(' B = ',D:3,M:3,S:5:1,' ' :3);
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
WRITELN (' ' :5,' h m s ', ' ' :10,' o ' ' ' ');
```

```
DMS(L/15,D,M,S); WRITE(' RA = ',D:2,M:3,S:5:1,' ' :3);
```

```
DMS(B,D,M,S); WRITE(' DEC = ',D:3,M:3,S:5:1,' ' :3);
```

```
END;
```

```
WRITELN (' R = ',R:12:8); WRITELN; WRITELN;
```

```
END; (* RESULT *)
```

```
(*-----*)
```

```
BEGIN (* COCO *)
```

```
GETINP (X,Y,Z,TEQX,ECLIPT);
```

```
RESULT (X,Y,Z,ECLIPT);
```

```
REPEAT
```

```
WRITE (' Command (?=Help): '); READLN (MODE); WRITELN;
```

```
IF MODE IN ['?', 'P', 'p', 'E', 'e', 'H', 'h', 'G', 'g'] THEN
```

```
CASE MODE OF
```

```
'?' : BEGIN (* help *)
```

```
WRITELN;
```

```
WRITELN (' E: equatorial <-> ecliptic ');
```

```
WRITELN (' P: -> precession G: -> geocentric ');
```

```
WRITELN (' H: -> heliocentric S: -> STOP ');
```

```
WRITELN;
```

```
END;
```

```
'P', 'p': BEGIN (* precession *)
```

```
WRITE (' New');
```

```
GETEQX(TEQX); (* read new equinox *)
```

```
IF ECLIPT THEN PMATECL(TEQX,TEQX,A)
```

```
ELSE PMATEQU(TEQX,TEQX,A);
```

```
PRECART(A,X,Y,Z);
```

```
TEQX := TEQX;
```

```
WRITELN;
```

```
WRITELN (' Coordinates referred to equinox T =',
```

```

                                TEQX:13:10);
END;

'E','e': BEGIN (* ecliptic <-> equatorial *)
    WRITELN;
    IF (ECLIPT) THEN
        BEGIN ECLEQU(TEQX,X,Y,Z); WRITE(' Equatorial'); END
    ELSE
        BEGIN EQUACL(TEQX,X,Y,Z); WRITE(' Ecliptic'); END;
    WRITELN (' coordinates: ');
    ECLIPT := NOT ECLIPT;
END;

'G','g', (* -> geocentric coordinates *)
'H','h': (* -> heliocentric coordinates *)
BEGIN
    GETDAT(T); (* read date *)
    SUN200(T,LS,BS,RS);
    CART(RS,BS,LS,XS,YS,ZS);
    PMATECL(T,TEQX,A);
    PRECART(A,XS,YS,ZS);
    IF NOT ECLIPT THEN ECLEQU(TEQX,XS,YS,ZS);
    IF MODE IN ['G','g']
        THEN (* -> geocentric *)
            BEGIN
                X:=X+XS; Y:=Y+YS; Z:=Z+ZS;
                WRITELN(' Geocentric coordinates: ');
            END
        ELSE (* -> heliocentric *)
            BEGIN
                X:=X-XS; Y:=Y-YS; Z:=Z-ZS;
                WRITELN(' Heliocentric coordinates: ');
            END;
END;

END;

IF NOT (MODE IN ['?', 'S', 's']) THEN RESULT(X,Y,Z,ECLIPT);

UNTIL MODE IN ['S', 's'];

END. (* COCO *)

(*-----*)

```

A simple menu enables the individual functions to be chosen. After entering the ecliptic or equatorial coordinates of a point, various commands allow any required coordinate system to be selected:

- E Conversion between ecliptic and equatorial coordinates,
- P Precession (Choice of equinox),
- G Conversion to geocentric coordinates,

H Conversion to heliocentric coordinates,

S STOP (Quit program).

The following example should illustrate the use and possibilities offered by COCO. The coordinates to be converted may be entered in either the equatorial or the ecliptic form. One can also choose between Cartesian coordinates or polar coordinates. This choice is offered by COCO at the beginning. All entries are indicated in *italic*.

COCO: coordinate conversion
version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Coordinate input: please select format required

1	ecliptic	cartesian	2	ecliptic	polar
3	equatorial	cartesian	4	equatorial	polar

4

We will choose equatorial polar coordinates, and enter the coordinates of the vernal equinox for epoch 1950. The distance is arbitrarily set at 1 AU. COCO accepts the initial entry and displays the data in Cartesian and polar coordinates.

Coordinates (RA (h m s) DEC (° ' ") R) ? *0 0 0.0 0 0 0.0 1.0*
equinox (yyyy) ? *1950.0*

(x,y,z) = (*1.00000000, 0.00000000, 0.00000000***)**

	h	m	s		°	'	"	
RA =	<i>0</i>	<i>0</i>	<i>0.0</i>	DEC =	<i>0</i>	<i>0</i>	<i>0.0</i>	R = <i>1.00000000</i>

We first want to convert the given position to epoch 2000, and therefore choose option P to calculate the precession. After entering the year, we obtain the position of the vernal equinox for 1950 referred to the new epoch 2000. The precession amounts to slightly more than half a degree.

Command (?=Help): *P*
New equinox (yyyy) ? *2000.0*

Coordinates for equinox T = 0.0000000000

(x,y,z) = (*0.99992571, 0.01117889, 0.00485898***)**

	h	m	s		°	'	"	
RA =	<i>0</i>	<i>2</i>	<i>33.7</i>	DEC =	<i>0</i>	<i>16</i>	<i>42.2</i>	R = <i>1.00000000</i>

We now need the given equatorial coordinates to be converted into the ecliptic system. We therefore choose option E and obtain the converted coordinates—again in Cartesian and polar forms. Instead of right ascension RA and

declination DEC we now have ecliptic longitude L and latitude B. As in the previous step, the distance remains unchanged.

Command (?=Help): *E*

Ecliptic coordinates:

(x,y,z) = (0.99992571, 0.01218922, 0.00001132)

$\begin{matrix} & ^{\circ} & ' & '' \\ L = & 0 & 41 & 54.3 \end{matrix}$
 $\begin{matrix} & ^{\circ} & ' & '' \\ B = & 0 & 0 & 2.3 \end{matrix}$
 $R = 1.00000000$

It is now possible to convert the given coordinates from geocentric to heliocentric coordinates using option H. COCO does not check in any way to determine in which of these two systems the actual data are expressed. It is therefore possible to call options H or G several times one after the other, but the results will be meaningless! The user is responsible for checking this particular aspect of the program. When the 'geocentric→heliocentric' (H) conversion or its converse (G) is selected, the date and time must be entered. COCO answers with the appropriate Julian Date and the converted data in the current coordinate system. As in the previous step we chose ecliptic coordinates, we now obtain heliocentric ecliptic coordinates. It will be noted that the distance has changed.

Command (?=Help):

H

Date (year month day hour) ? 1989 1 1 0.0

JD 2447527.5000

Heliocentric coordinates:

(x,y,z) = (0.81725249, 0.97838163, 0.00003596)

$\begin{matrix} & ^{\circ} & ' & '' \\ L = & 50 & 7 & 39.5 \end{matrix}$
 $\begin{matrix} & ^{\circ} & ' & '' \\ B = & 0 & 0 & 5.8 \end{matrix}$
 $R = 1.27480675$

The conversion into ecliptic coordinates can be reversed by selecting option E again, which converts the current (ecliptic) coordinates into equatorial coordinates.

Command (?=Help): *E*

Equatorial coordinates:

(x,y,z) = (0.81725249, 0.89763329, 0.38921086)

$\begin{matrix} & h & m & s \\ RA = & 3 & 10 & 44.1 \end{matrix}$
 $\begin{matrix} & ^{\circ} & ' & '' \\ DEC = & 17 & 46 & 36.5 \end{matrix}$
 $R = 1.27480675$

Let us now recalculate the coordinates for the original equinox of 1950.

Command (?=Help): *P*

New equinox (yyyy) ? *1950.0*

Coordinates referred to equinox T = -0.5000000000

(x,y,z) = (0.82911749, 0.88843066, 0.38521086)

	h	m	s		o	'	"		
RA =	3	7	54.7	DEC =	17	35	17.2	R =	1.27480675

Conversion of these coordinates into geocentric coordinates should now yield the original coordinates of the vernal equinox, because we have applied inverse transformation to the three conversions that we invoked. This assumes that the same date and time are entered as were used previously in calculating the heliocentric coordinates. Unavoidable rounding errors within the computer will almost always prevent exactly the same coordinates as those entered at the beginning of this example from being obtained.

Command (?=Help): *G*

Date (year month day hour) ? *1989 1 1 0.0*

JD 2447527.5000

Geocentric coordinates:

(x,y,z) = (1.00000000, 0.00000000, 0.00000000)

	h	m	s		o	'	"		
RA =	23	59	60.0	DEC =	0	0	0.0	R =	1.00000000

Entering S exits the program.

Command (?=Help): *S*

The values shown here may differ slightly depending on the particular Pascal compiler being used.

3. Calculation of Rising and Setting Times

3.1 The Observer's Horizon System

The ecliptic and equatorial coordinates discussed so far are specified in terms of the mean plane of the Earth's orbit and of the position of the Earth's axis. Neither of these systems is particularly suitable, however, for an observer situated on the surface of the Earth. As such an observer (without being aware of it) takes part in the Earth's daily rotation, it appears as if the Sun, Moon, and stars follow large arcs across the sky from East to West during the course of a day, reaching their highest point above the horizon when they are on the meridian.

The stars' apparent paths depend on the geographical latitude of the point of observation. In the Southern Hemisphere most stars do not reach their greatest altitude in the South, but in the North. As a result, orienting themselves on the sky is more difficult for observers who are used to the appearance of the sky at moderate northern latitudes. All the well-known constellations appear to be standing on their heads.

Two points on the celestial sphere are particularly significant: the zenith (the point directly above the observer), and the North Celestial Pole (Fig. 3.1). The latter is taken to be the point on the celestial sphere towards which the Earth's axis is pointing, and about which, in consequence, all the stars appear to be moving in concentric circles. The altitude of this point above the horizon corresponds to the observer's geographic latitude φ . A great circle drawn through the North Celestial Pole and the zenith, known as the *meridian*, intersects the horizon at the exact North and South points.

The coordinates used in the horizontal system are azimuth (A) and altitude (h), which may, for instance, be determined by the use of a theodolite (see Fig. 3.1). The altitude is simply the angular height above the horizon. The azimuth, determined by rotating the theodolite around its vertical axis, is defined as the angle between the South point and the required position. On this basis, the azimuth of the West point is $A = 90^\circ$ and, correspondingly, for the East point $A = 270^\circ$ (or $A = -90^\circ$). Unfortunately, alongside the definition just given, there is a second one that is used, in particular, in navigation. Here, azimuth is taken to originate at the North point. According to this method, the East point therefore has an azimuth of 90° . In case of any doubt, it is essential to check which of these two definitions of azimuth is being used!

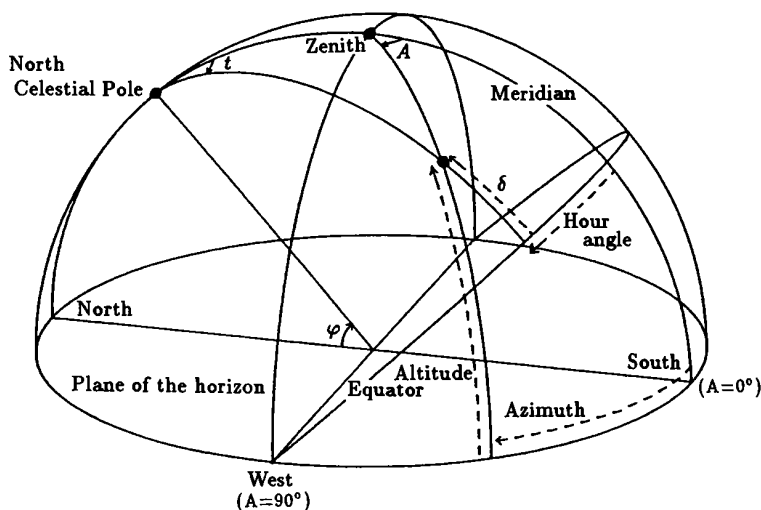


Fig. 3.1. The horizontal system

Because of the rotation of the Earth, the altitude and azimuth of an object with a given right ascension and declination alter continuously. If a telescope is held at a fixed position in the horizontal system, with time, the stars that cross the field of view are those that have the same declination, and which differ only in their right ascension. The determination of the declination of a star from altitude and azimuth is therefore independent of time. For right ascension, on the other hand, only the difference between the object and stars on the meridian can be determined.

In order to obtain reciprocal conversions between horizontal and equatorial coordinates, the concept of *hour angle* τ is used. This is the difference between the right ascension of the star being observed and the right ascension of stars on the meridian. Hour angle τ , like right ascension α , is generally measured in units of time ($1^h \equiv 15^\circ$), and therefore corresponds approximately¹ to the amount of time that has passed since the star crossed the meridian. In the system employed here, A and τ have the same mathematical sign. If we have the Cartesian coordinates

$$\begin{aligned}
 x &= \cos h \cos A & \hat{x} &= \cos \delta \cos \tau \\
 y &= \cos h \sin A & \hat{y} &= \cos \delta \sin \tau \\
 z &= \sin h & \hat{z} &= \sin \delta
 \end{aligned}
 \tag{3.1}$$

¹One rotation of the Earth takes only 23^h56^m .

the following transformations apply:

$$\begin{aligned} x &= +\hat{x} \sin \varphi - \hat{z} \cos \varphi & \hat{x} &= +x \sin \varphi + z \cos \varphi \\ y &= +\hat{y} & \hat{y} &= +y \\ z &= +\hat{x} \cos \varphi + \hat{z} \sin \varphi & \hat{z} &= -x \cos \varphi + z \sin \varphi \end{aligned} \quad (3.2)$$

Here φ is the geographical latitude of the observer, and $\hat{}$ indicates equatorial coordinates. Substitution then yields a system of three equations

$$\begin{aligned} \cos h \cos A &= +\cos \delta \cos \tau \sin \varphi - \sin \delta \cos \varphi \\ \cos h \sin A &= +\cos \delta \sin \tau \\ \sin h &= +\cos \delta \cos \tau \cos \varphi + \sin \delta \sin \varphi \end{aligned} \quad (3.3)$$

for computing azimuth and altitude from hour angle and declination. Conversely, the hour angle and declination may be determined from

$$\begin{aligned} \cos \delta \cos \tau &= +\cos h \cos A \sin \varphi + \sin h \cos \varphi \\ \cos \delta \sin \tau &= +\cos h \sin A \\ \sin \delta &= -\cos h \cos A \cos \varphi + \sin h \sin \varphi \end{aligned} \quad (3.4)$$

for a given azimuth and altitude. The transformation between the equatorial system and the horizontal system may be accomplished by the sub-routines EQUHOR and HOREQU, which are given here for the sake of completeness.

```
(*-----*)
(* EQUHOR: conversion of equatorial into horizontal coordinates *)
(* DEC : declination (-90 deg .. +90 deg) *)
(* TAU : hour angle (0 deg .. 360 deg) *)
(* PHI : geographical latitude (in deg) *)
(* H : altitude (in deg) *)
(* AZ : azimuth (0 deg .. 360 deg, counted S->W->N->E->S) *)
(*-----*)
```

```
PROCEDURE EQUHOR (DEC,TAU,PHI: REAL; VAR H,AZ: REAL);
  VAR CS_PHI,SN_PHI, CS_DEC,SN_DEC, CS_TAU, X,Y,Z, DUMMY: REAL;
  BEGIN
    CS_PHI:=CS(PHI); SN_PHI:=SN(PHI);
    CS_DEC:=CS(DEC); SN_DEC:=SN(DEC); CS_TAU:=CS(TAU);
    X:=CS_DEC*SN_PHI*CS_TAU - SN_DEC*CS_PHI;
    Y:=CS_DEC*SN(TAU);
    Z:=CS_DEC*CS_PHI*CS_TAU + SN_DEC*SN_PHI;
    POLAR (X,Y,Z, DUMMY,H,AZ)
  END;
```

```
(*-----*)
(* HOREQU: conversion of horizontal to equatorial coordinates *)
(* H,AZ : azimuth and altitude (in deg) *)
(* PHI : geographical latitude (in deg) *)
(* DEC : declination (-90 deg .. +90 deg) *)
(* TAU : hour angle (0 deg .. 360 deg) *)
(*-----*)
```

```
PROCEDURE HOREQU (H,AZ,PHI: REAL; VAR DEC,TAU: REAL);
  VAR CS_PHI,SN_PHI, CS_H,SN_H, CS_AZ, X,Y,Z, DUMMY: REAL;
```

```

BEGIN
  CS_PHI:=CS(PHI); SN_PHI:=SN(PHI);
  CS_H :=CS(H);   SN_H :=SN(H); CS_AZ:=CS(AZ);
  X := CS_H*SN_PHI*CS_AZ+SN_H*CS_PHI;
  Y := CS_H*SN(AZ);
  Z := SN_H*SN_PHI-CS_H*CS_PHI*CS_AZ;
  POLAR (X,Y,Z, DUMMY,DEC,TAU)
END;
(*-----*)

```

For determining the times of rising and setting, which is the aim of this chapter, only one equation is required, namely

$$\sin h = \cos \varphi \cos \delta \cos \tau + \sin \varphi \sin \delta \quad (3.5)$$

It allows the altitude h to be calculated from given values of geographical latitude φ , declination δ , and hour angle τ .

Before we can carry out this conversion, however, some preparation is required. First, we lack the means of determining the coordinates (α, δ) of the Sun and the Moon. Then we need to clarify how we can calculate the hour angle from the known right ascension at a particular time. Finally, we have to take into account a whole series of corrections that affect the height of the observed horizon. Only after these steps will we return to the equation just mentioned.

3.2 Sun and Moon

Calculation of rising and setting does not make demands for a high degree of accuracy in the coordinates of the Sun and the Moon. The **MINI_SUN** and **MINI_MOON** procedures therefore contain only the most important terms describing the respective orbits. They are greatly reduced versions of **SUN200** and **MOON**, which will be described in detail in Chap. 6 and Chap. 8. Conversion of ecliptic longitude and latitude into equatorial coordinates is also included, therefore both procedures can be used without further operations to obtain the right ascension and declination of the object in question.

```

(*-----*)
(* MINI_MOON: low precision lunar coordinates (approx. 5'/1') *)
(*      T : time in Julian centuries since J2000 *)
(*      ( T=(JD-2451545)/36525 ) *)
(*      RA : right ascension (in h; equinox of date) *)
(*      DEC: declination (in deg; equinox of date) *)
(*-----*)
PROCEDURE MINI_MOON (T: REAL; VAR RA,DEC: REAL);
  CONST P2 =6.283185307; ARC=206264.8062;
  COSEPS=0.91748; SINEPS=0.39778; (* cos/sin(obliquity ecliptic) *)
  VAR  L0,L,LS,F,D,H,S,W,DL,CB : REAL;
  L_MOON,B_MOON,V,W,X,Y,Z,RHO: REAL;

```



```

FUNCTION FRAC(X:REAL):REAL;
  (* with some compilers it may be necessary to replace *)
  (* TRUNC by LONG_TRUNC oder INT if T<-24! *)
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;
BEGIN
  (* mean elements of lunar orbit *)
  LO:= FRAC(0.606433+1336.855225*T); (* mean longitude Moon (in rev) *)
  L :=P2*FRAC(0.374897+1325.552410*T); (* mean anomaly of the Moon *)
  LS:=P2*FRAC(0.993133+ 99.997361*T); (* mean anomaly of the Sun *)
  D :=P2*FRAC(0.827361+1236.853086*T); (* diff. longitude Moon-Sun *)
  F :=P2*FRAC(0.259086+1342.227825*T); (* mean argument of latitude *)
  DL := +22640*SIN(L) - 4586*SIN(L-2*D) + 2370*SIN(2*D) + 769*SIN(2*L)
        -668*SIN(LS)- 412*SIN(2*F) - 212*SIN(2*L-2*D) - 206*SIN(L+LS-2*D)
        +192*SIN(L+2*D) - 165*SIN(LS-2*D) - 125*SIN(D) - 110*SIN(L+LS)
        +148*SIN(L-LS) - 55*SIN(2*F-2*D);
  S := F + (DL+412*SIN(2*F)+541*SIN(LS)) / ARC;
  H := F-2*D;
  N := -526*SIN(H) + 44*SIN(L+H) - 31*SIN(-L+H) - 23*SIN(LS+H)
        + 11*SIN(-LS+H) -25*SIN(-2*L+F) + 21*SIN(-L+F);
  L_MOON := P2 * FRAC ( LO + DL/1296E3 ); (* in rad *)
  B_MOON := ( 18520.0*SIN(S) + N ) / ARC; (* in rad *)
  (* equatorial coordinates *)
  CB:=COS(B_MOON);
  X:=CB*COS(L_MOON); V:=CB*SIN(L_MOON); W:=SIN(B_MOON);
  Y:=COSEPS*V-SINEPS*W; Z:=SINEPS*V+COSEPS*W; RHO:=SQRT(1.0-Z*Z);
  DEC := (360.0/P2)*ARCTAN(Z/RHO);
  RA := ( 48.0/P2)*ARCTAN(Y/(X+RHO)); IF RA<0 THEN RA:=RA+24.0;
END;

(*-----*)
(* MINI_SUN: low precision solar coordinates (approx. 1') *)
(* T : time in Julian centuries since J2000 *)
(* ( T=(JD-2451545)/36525 ) *)
(* RA : right ascension (in h; equinox of date) *)
(* DEC: declination (in deg; equinox of date) *)
(*-----*)

PROCEDURE MINI_SUN(T:REAL; VAR RA,DEC: REAL);
CONST P2 = 6.283185307; COSEPS=0.91748; SINEPS=0.39778;
VAR L,M,DL,SL,X,Y,Z,RHO: REAL;
FUNCTION FRAC(X:REAL):REAL;
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;
BEGIN
  M := P2*FRAC(0.993133+99.997361*T);
  DL := 6893.0*SIN(M)+72.0*SIN(2*M);
  L := P2*FRAC(0.7859453 + M/P2 + (6191.2*T+OL)/1296E3);
  SL := SIN(L);
  X:=COS(L); Y:=COSEPS*SL; Z:=SINEPS*SL; RHO:=SQRT(1.0-Z*Z);
  DEC := (360.0/P2)*ARCTAN(Z/RHO);
  RA := ( 48.0/P2)*ARCTAN(Y/(X+RHO)); IF (RA<0) THEN RA:=RA+24.0;
END;
(*-----*)

```

3.3 Sidereal Time and Hour Angle

The right ascensions of the Sun and Moon are not sufficient to calculate their altitudes above the horizon. We also need to know the right ascension of stars on the meridian and, from that, determine the hour angle. Precisely which stars these are depends on the observer's location and the exact time. As the Earth rotates around its axis once a day, an observer sees all right ascensions between 0^h and 24^h transit the meridian. Over an hour, the right ascension of the stars that culminate alters by about 1^h .

The regular motion of the stars is very suitable for forming the basis for another system of time reckoning, which is known as *sidereal time*. For any given place, it is defined as the right ascension of the stars that are on the meridian at that particular instant. Because of this definition, sidereal time can be determined directly by observation of the sky.

The need for sidereal time, in addition to solar time, arises from the slight difference between the length of a solar day and one rotation of the Earth. Clocks, which we use continually to tell the time, are arranged to divide the day into 24 hours. One day is therefore the alternation of light and dark, which is determined by the Sun, and 24 hours are the amount of time, on average, between one meridian passage of the Sun and the next. If, however, we measure the corresponding interval for a star, we find that this takes only $23^h56^m4^s.091$. This shorter period, known as a sidereal day in distinction to a solar day, is exactly the duration of one rotation of the Earth. The cause of the difference of about 4^m arises from the Earth's year-long orbit around the Sun. Because of this motion, the Sun's right ascension changes by $360^\circ \equiv 24^h$ per year, which is about 4^m per day. So the time between two meridian transits of the Sun is greater than that for a star by this amount.

For any given Universal Time UT, the sidereal time at Greenwich may be determined from

$$\begin{aligned} \Theta_0 = & 24110^s54841 + 8640184^s812866 \cdot T_0 + 1.0027379093 \cdot UT \\ & + 0^s.093104 \cdot T^2 - 0^s.0000062 \cdot T^3 \end{aligned} \quad (3.6)$$

where

$$T_0 = \frac{JD_0 - 2451545}{36525} \quad \text{and} \quad T = \frac{JD - 2451545}{36525} .$$

JD and JD_0 are the Julian Date of the time of observation, and the Julian Date at 0^h UT on the date of observation. For a position with a geographical longitude λ , the local sidereal time differs by $\lambda/15^\circ$ hours from the sidereal time at Greenwich:

$$\Theta = \Theta_0 - \lambda \cdot 1^h/15^\circ . \quad (3.7)$$

Here, longitude λ is reckoned *positive towards the west* (for Munich in Germany, for example, $\lambda = -11.6$). The hour angle τ of a star of right ascension α is therefore given by

$$\tau = \Theta - \alpha \quad . \quad (3.8)$$

```
(*-----*)
(* LMST: local mean sidereal time *)
(*-----*)
FUNCTION LMST(MJD,LAMBDA:REAL):REAL;
  VAR MJDO,T,UT,GMST: REAL;
  FUNCTION FRAC(X:REAL):REAL;
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;
  BEGIN
    (* MJDO:=TRUNC(MJD); *)      (* Standard Pascal *)
    MJDO:=INT(MJD);              (* TURBO Pascal *)
    (* MJDO:=LONG_TRUNC(MJD); *) (* ST Pascal plus *)
    UT:=(MJD-MJDO)*24; T:=(MJDO-51544.5)/36525.0;
    GMST:=6.697374558 + 1.0027379093*UT
           + (8640184.812866+(0.093104-6.2E-6*T)*T)*T/3600.0;
    LMST:=24.0*FRAC( (GMST-LAMBDA/15.0) / 24.0 );
  END;
(*-----*)
```

As an example for the use of sidereal time in the conversion of right ascension and hour angle, the following sample program computes the azimuth and altitude of the star Deneb (α Cyg) for a given place and time:

```
VAR LAMBDA, PHI, RA, DELTA, MODJD, TAU,H, A: REAL;
BEGIN
  LAMBDA:=-11.6; PHI:=48.1;      (* observing site Munich *)
  DDD (20,41,12.8, RA);          (* equatorial coordinates of Deneb *)
  DDD (45,15,25.8, DELTA);
  MODJD := MJD (1,8,1993,21.0);  (* Date: Aug 1, 1993, 21:00 UT *)
  TAU := 15.0 * (LMST(MODJD,LAMBDA) - RA);
  EQUHOR ( DELTA,TAU,PHI, H,A );
  WRITELN ('DENEb:');
  WRITELN ('altitude ','H:7:2,' deg');
  WRITELN ('azimuth ','A:7:2,' deg');
END.
```

3.4 Universal Time and Ephemeris Time

We have already frequently used the concept of "Time", without considering it in more detail. In astronomy, however, we encounter a whole series of ways of reckoning time that are employed alongside one another. In general, they may be divided into two classes, the most important members of which are

dynamical time (TT, TDB) and *Universal Time* (UT). The different objectives that are the reasons for these different times need to be explained in moderate detail.

The concept behind dynamical time is that it is a time that allows astronomical processes to be described in terms of physics. One second of dynamical time is defined by the period of a specific transition occurring in the caesium atom, and is correspondingly measured nowadays by atomic clocks. Since 1984, dynamical time has replaced *Ephemeris Time* (ET). The basis on which the latter had previously been established was that of tables of the motion of the Sun, the Moon, and the planets, which had been calculated according to classical mechanics. Ephemeris Time could be determined by comparing the observed positions of these celestial bodies with the ephemerides calculated earlier. The grounds for the introduction of dynamical time lay in relativity theory, which established that there is no truly universally applicable time, but that its rate also depends on the position and motion of the reference system in which time is being measured. This necessitates, in particular, differentiating between Terrestrial (Dynamical) Time (TT or TDT), which refers to the centre of the Earth, and Barycentric Dynamical Time (TDB), which relates to the centre of gravity of the Solar System. For our purposes, however, the three forms of time ET, TT, and TDB may be regarded as equivalent.

In contrast to Ephemeris Time and dynamical time, *Universal Time* (UT) is a *non-uniform* timescale. UT is the best current realization of a solar time. It was established to try to ensure that over several thousand years one day has an average length of 24 hours. But the result of this is that the length of one second of Universal Time is not constant, because the actual mean length of a day depends on the rotation of the Earth and the apparent motion of the Sun (i.e., the length of the year). Unfortunately, it is not possible to determine Universal Time by a suitable conversion from dynamical time, because the rotation of the Earth cannot be predicted accurately. Every change in the Earth's rotation alters the length of the day, and must therefore be taken into account in UT. Universal Time is therefore defined as a function of sidereal time, which directly reflects the rotation of the Earth. For any particular day, 0^h UT is defined as the instant at which Greenwich Mean Sidereal Time (GMST) has the value

$$\begin{aligned} \text{GMST}(0^h \text{UT}) = & 24110^{\circ}54841 + 8640184^{\circ}812866 \cdot T_0 \\ & + 0^{\circ}093104 \cdot T_0^2 - 0^{\circ}0000062 \cdot T_0^3 \end{aligned}$$

where

$$T_0 = \frac{\text{JD}(0^h \text{UT}) - 2451545}{36525}.$$

We have already encountered this equation in Sect. 3.3 where we used it to calculate sidereal time for a given instant of Universal Time. We have now,

however, established that sidereal time is the only observable quantity from which we can derive Universal Time.

The difference between Universal Time and Ephemeris Time or Terrestrial Time can only be determined retrospectively. Table 3.1 summarizes the value of $\Delta T = ET - UT$ ($ET = TT = TDB$) over the course of this century. At present ΔT is increasing by about 0.5 to 1.0 seconds per year.

Table 3.1. Value of $\Delta T = ET - UT$ (in s) for 1900-1990

Year	ET-UT	Year	ET-UT	Year	ET-UT	Year	ET-UT
1900	-2.72	1925	23.62	1950	29.15	1975	45.48
1905	3.86	1930	24.02	1955	31.07	1980	50.54
1910	10.46	1935	23.93	1960	33.15	1985	54.34
1915	17.20	1940	24.33	1965	35.73	1990	56.86
1920	21.16	1945	26.77	1970	40.18		

Clock time, which we use for everyday purposes, is derived from *Coordinated Universal Time* (UTC). UTC is obtained from atomic clocks, and therefore has the same rate as Terrestrial Time. By the use of leap seconds, which may be inserted once or twice a year, care is taken to ensure that UTC never deviates more than 0.9 seconds from Universal Time UT. UTC is the basis for time measurement over the whole of the Earth. Every point is part of a specific time zone, within which the official time differs by whole hours (or half hours) from UTC. In practice, the time zones have been arranged to agree with geographical features and, in particular, with national borders, in order to avoid having several time zones across a single country. The time zone employed in a particular area is established by international agreement and may be determined from a chart of the time zones.

At this point we may well enquire which time we should use when we want to calculate the rising and setting times of the Sun and the Moon. The preceding definitions give us the following general rule of thumb for the use of Universal Time and Ephemeris Time:

- Universal Time (UT) is used to calculate sidereal time.
- Ephemeris Time (ET) or dynamical time (TT, TDB) are used to calculate solar, lunar, or planetary ephemerides.

Let us take as an example the calculation of the altitude of the Moon, and consider the individual steps in the process. Let us take the time as being 1982 January 1, 0^h Central European Time, and the observing site as Munich ($\varphi = 48^\circ 1'$, $\lambda = -11^\circ 6'$). We can ignore the slight difference between UTC and UT, so we obtain the Universal Time from $UT = CET - 1^h$. As discussed earlier, Universal Time has to be used in calculating sidereal time, while Ephemeris Time is used in calculating the coordinates of the Moon. We obtain the conversion as shown in this portion of a program:

```

CONST DAY=1; MONTH=1; YEAR=1982; CET=0.0;      (* Date *)
UT_MINUS_CET = -1.0;                          (* Zone time difference *)
ET_MINUS_UT = 52.17;                          (* in sec; for 1982 *)
PHI=48.1; LAMBDA=-11.6;                       (* geogr. coordinates *)

VAR UT,MJD_UT,MJD_ET,T_UT,T_ET : REAL;
    RA,DEC,THETA,TAU,SIN_H : REAL;

BEGIN

    UT := CET + UT_MINUS_CET;
    MJD_UT := MJD(DAY,MONTH,YEAR,UT);          (* MJD Universal Time *)
    MJD_ET := MJD_UT + (ET_MINUS_UT/86400.0);  (* MJD Ephemeris Time *)

    T_ET := (MJD_ET-51544.5)/36525.0;          (* centuries since J2000 *)
    MINI_MOON(T_ET,RA,DEC);                   (* equatorial coordinates *)

    THETA := LMST(MJD_UT,LAMBDA);              (* loc.sidereal time (h) *)
    TAU := 15.0 * (THETA-RA);                  (* hour angle (h) *)

    SIN_H :=                                  (* sine of altitude *)
        SN(PHI)*SN(DEC)+CS(PHI)*CS(DEC)*CS(TAU); (* above the horizon *)

END.
```

We can also ask what errors occur if we ignore the difference between Universal Time and Ephemeris Time. If the coordinates of the Moon are calculated with the simplification that we use UT instead of ET, the example just given becomes:

```

UT := CET + UT_MINUS_CET;
MJD_UT := MJD(DAY,MONTH,YEAR,UT);          (* MJD Universal Time *)

T_UT := (MJD_UT-51544.5)/36525.0;          (* centuries since J2000 *)
MINI_MOON(T_UT,RA,DEC);                   (* equatorial coordinates *)

THETA := LMST(MJD_UT,LAMBDA);              (* loc.sidereal time (h) *)
TAU := 15.0 * (THETA-RA);                  (* hour angle (h) *)

SIN_H :=                                  (* sine of altitude *)
    SN(PHI)*SN(DEC)+CS(PHI)*CS(DEC)*CS(TAU); (* above the horizon *)
```

Although the sidereal time is calculated correctly, the coordinates of the Moon are now calculated for a time that is $\Delta T = ET - UT$ too *early*. At present, ΔT is about one minute, an interval of time in which the Moon moves by about 30 arc-seconds. This error is even smaller than the error obtained by using the simplified MINI_MOON procedure. The times of moonrise and moonset will be in error by only about 3". For the Sun, these values are even smaller. In the SUNSET program, we can therefore ignore ΔT with a clear conscience. In later applications, such as the calculation of exact planetary positions or in predicting stellar occultations, it is, however, essential to distinguish carefully

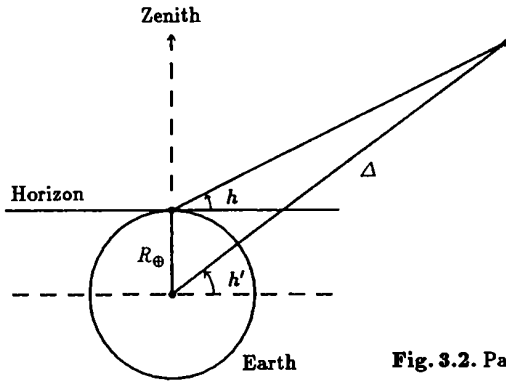


Fig. 3.2. Parallax

between the various forms of time. This is why we have discussed in detail the various methods of reckoning astronomical time.

3.5 Parallax and Refraction

So far, we have always given the right ascension and declination of a celestial body in geocentric equatorial coordinates, i.e., in a system with its origin at the centre of the Earth. We are not at the centre of the Earth, however, and observe from its surface. What differences does this cause between the coordinates that are calculated and those actually observed?

In observing fixed stars, in practice no difference is found between geocentric coordinates and the *topocentric* coordinates that are obtained by parallel translation of the zero point of the coordinates from the centre of the Earth to the position of the observer on the surface. The distance of the fixed stars is enormous in comparison with the distance between the observer and the centre of the Earth. But when we are dealing with nearer celestial bodies, such as the planets, the Sun, or the Moon, there are marked differences.

This difference is known as *parallax* and causes an object observed from the surface of the Earth to appear slightly lower than it would if it were observable from the centre of the Earth. The topocentric altitude h is less than the geocentric altitude h' (see Fig. 3.2). If the object is at the zenith, the parallax disappears. As the altitude decreases, however, the parallax becomes increasingly important, and becomes a maximum when the altitude $h = 0^\circ$. This is known as the *horizontal parallax* and is calculated from

$$\pi = \arcsin(R_\oplus/\Delta) \quad , \quad (3.9)$$

where $R_\oplus \approx 6378$ km, the distance of the observer from the centre of the Earth, and Δ the *geocentric* distance of the object. If we substitute the distance of the Sun in (3.9) then we obtain a value of $\pi_\odot = 8''.8$. For the Moon,

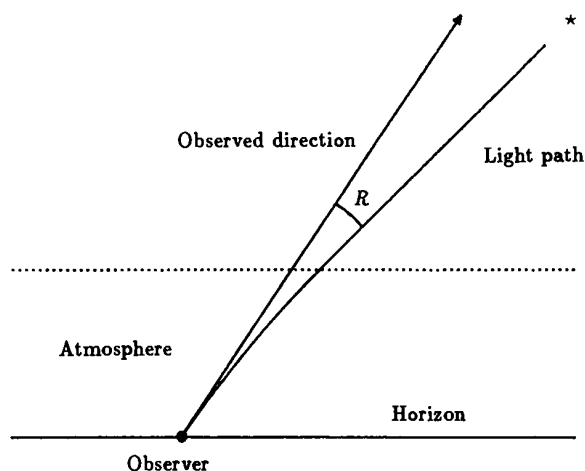


Fig. 3.3. Refraction

our nearest neighbour, on the other hand, the horizontal parallax amounts to the no longer insignificant amount of

$$\pi_{\text{Moon}} = \arcsin \left(\frac{6378 \text{ km}}{384400 \text{ km}} \right) \approx 57' .$$

We can already see that for calculating rising and setting times we must use the topocentric altitude rather than the geocentric one. Before that, however, yet another effect must be described.

In passing from the vacuum of space into the optically denser atmosphere of the Earth, rays of light are deviated, in accordance with the law of refraction, towards the vertical (see Fig. 3.3). This is known as *atmospheric refraction*. To an observer on the ground, stars will appear to have been 'raised' slightly. A ray of light entering the atmosphere at a low angle has a longer path through layers of air of differing densities (and refractive indexes) than a ray entering vertically. Atmospheric refraction is therefore greater, the lower the altitude of the object. The maximum value of about 34 arc-minutes is attained at the horizon. Some further data are given in Table 3.2. In order to obtain the observed altitude, R should be added to the calculated topocentric altitude. Atmospheric refraction also depends on the density and temperature of the air. The figures quoted should therefore be taken as average values. Comprehensive tables of normal refraction and numerical approximation formulae can be found in appropriate reference works.

Table 3.2. Values of refraction near the horizon

h	10°	5°	2°	1°	0°
R	$5'31''$	$10'15''$	$19'7''$	$25'36''$	$34'$

We have seen that when dealing with rising and setting times, we must use topocentric altitudes. In addition, we also have to take refraction near the horizon into account. Because rising and setting times are always referred to the upper limb of the Sun or the Moon, we also need to make allowance for the apparent radius $s_{\odot}$ or s_{Moon} . At the instant the body rises or sets, it therefore has the *geocentric* altitude

$$h_{R/S} = 0^\circ + \pi - R_{h=0} - s. \quad (3.10)$$

Our task now consists of finding, for a given date, the time at which the body being observed reaches an altitude of $h = h_{R/S}$. Because there is no sense in trying to obtain the times to an accuracy better than a few minutes, we can forgo accurate calculation of π and s and use average values. The values normally taken in calculating rising and setting times are therefore:

- Sunrise or sunset : $h_{R/S} = -0^\circ 50'$,
- Moonrise or moonset : $h_{R/S} = +0^\circ 08'$,
- for Stars or Planets : $h_{R/S} = -0^\circ 34'$.

As our program also calculates the times of the beginning and end of twilight, the necessary definitions should also be discussed here. The types of twilight are

- *astronomical* twilight when $h_{\odot} = -18^\circ$,
- *nautical* twilight when $h_{\odot} = -12^\circ$ and
- *civil* twilight when $h_{\odot} = -6^\circ$.

No further corrections are applied to these values for parallax, refraction, or apparent semi-diameter. Twilight therefore begins or ends, by definition, when the *geocentric* altitude of the Sun attains one of the values quoted.

We now only have to solve the general problem of finding the instant of time at which a celestial body reaches a predetermined altitude.

3.6 Rising and Setting Times

Equation (3.5) allows us to determine the altitude h , given the geographical latitude φ , the declination δ , and the hour angle τ . The hour angle is obtained for a given local sidereal time from (3.8).

If, on the other hand, we have the altitude h , we can rearrange (3.5), and obtain a relationship for the hour angle:

$$\cos \tau = \frac{\sin h - \sin \varphi \sin \delta}{\cos \varphi \cos \delta}. \quad (3.11)$$

We can therefore calculate the hour angle of a star when it is observed at an altitude h above the horizon. The equation is not, however, valid for every

possible value of h , but only for the altitudes that the star can actually attain. At a geographical latitude of $\varphi = 50^\circ$, a star of declination $\delta = 60^\circ$, for example, is circumpolar, and can only be observed between $h = 20^\circ$ and $h = 80^\circ$. If, for a predetermined altitude h , the value of $|\cos \tau|$ is consistently greater than 1, then the star will always be found either above or below that altitude.

For the rising and setting of stars, we substitute the altitude $h_{R/S} = -0^\circ 34'$ in (3.11), the hour angle thus obtained being expressed in units of time ($15^\circ \equiv 1^h$). It gives the number of hours in sidereal time that a star that is just rising requires to reach culmination. A sidereal day is equal to $23^h 56^m 4^s.091$ solar time, giving a factor of $23^h 56^m 4^s.091 / 24^h = 0.9972696$ by which an interval stated in sidereal time may be converted into solar time. If we multiply the hour angle by this factor, we obtain what is known as the *semi-diurnal arc*. This is half the overall time that the star is visible above the horizon.

For a given right ascension α_* of a star, the sidereal time at the instant of rising or setting is given by

$$\Theta_{R/S} = \begin{cases} \alpha_* - \tau & \text{for rise} \\ \alpha_* + \tau & \text{for set} \end{cases}.$$

If the Local Sidereal Time at 0^h is given by Θ_0 , then the star rises

$$0.9973 \cdot (\Theta_0 - \Theta_R)$$

before midnight, and sets

$$0.9973 \cdot (\Theta_S - \Theta_0)$$

after midnight.

In contrast to the coordinates of the stars, those of the Sun and the Moon alter noticeably during the course of a day. In order to calculate the hour angle and the times of rising and setting from the equations just given, however, we require the right ascension and declination at the moment the bodies cross the horizon. This requires an iterative procedure. The rising and setting times are first calculated for the coordinates at an arbitrary time for that particular day (in general, $t = 12^h$ is chosen). With the approximation thus obtained, improved coordinates can be determined, and these then again used to obtain more accurate rising and setting times. In the case of the Moon, these steps are repeated until the times obtained differ by less than one minute. For the Sun and the planets, the first iteration gives a sufficiently accurate result.

Unfortunately, there are a few problem cases, particularly in determining the times of moonrise and moonset, where the simple iteration method is difficult to apply. The following points need to be borne in mind:

- As the Moon orbits the Earth from West to East, it remains longer in the sky than stars with the same declination. As a result, moonrise

is, on average, about 50^m later each day. This means that there is one day in every month on which the Moon does not rise, and also another day on which the Moon does not set. To give an example: On 1988 February 8 at Munich, the Moon rose at 23^h30^m and set at 9^h41^m on the following day. Because of its motion relative to the stars, it then remained below the horizon until after midnight. So the next moonrise did not take place on February 9, but on February 10, at 0^h43^m .

- At high geographical latitudes, the Sun and the Moon frequently remain below the horizon for days. The rising and setting times are then primarily determined by the daily change in declination. Possible grazing contacts with the horizon are thus difficult to establish using the iteration method.

Because of this, the **SUNSET** program is based on another principle. The rising and setting times are determined by inverse interpolation of a series of solar and lunar altitudes. This does require a greater computational effort, but leads to a significant simplification of the program's structure.

3.7 Quadratic Interpolation

If we calculate a table of altitudes at hourly intervals, it is then possible to represent the behaviour of the altitudes as a function of time through a simple interpolation function. Quadratic interpolation is suitable for our purposes: from three values we may calculate the coefficients of a parabola, which approximates the behaviour of the function between the given points. We work from three values of the function:

$$y_- = f(x=-1), \quad y_0 = f(x=0) \quad \text{and} \quad y_+ = f(x=+1) \quad .$$

We require the coefficients a , b and c of a parabola

$$y = a \cdot x^2 + b \cdot x + c \quad , \quad (3.12)$$

which runs through the points $(-1, y_-)$, $(0, y_0)$ and $(1, y_+)$. Substituting into the formula for the parabola, we have the three equations

$$\begin{aligned} y_- &= a - b + c \\ y_0 &= c \\ y_+ &= a + b + c \quad , \end{aligned}$$

from which the required coefficients of the parabola may be obtained:

$$\begin{aligned} a &= (y_+ + y_-)/2 - y_0 \\ b &= (y_+ - y_-)/2 \\ c &= y_0 \quad . \end{aligned} \quad (3.13)$$

For $b^2 \geq 4ac$, this parabola is zero at the points:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (3.14)$$

and has an extreme value

$$\begin{aligned} x_E &= \frac{-b}{2a} \\ y_E &= a \cdot x_E^2 + b \cdot x_E + c \end{aligned} \quad (3.15)$$

The QUAD procedure solves these equations and also chooses just the zero points that lie between $x = -1$ and $x = +1$.

```
(*-----*)
(* QUAD: finds a parabola through 3 points *)
(*      (-1,Y_MINUS), (0,Y_0) und (1,Y_PLUS), *)
(*      that do not lie on a straight line. *)
(*-----*)
(*      Y_MINUS,Y_0,Y_PLUS: three y-values *)
(*      XE,YE      : x and y of the extreme value of the parabola *)
(*      ZERO1      : first root within [-1,+1] (for NZ=1,2) *)
(*      ZERO2      : second root within [-1,+1] (only for NZ=2) *)
(*      NZ         : number of roots within the interval [-1,+1] *)
(*-----*)
PROCEDURE QUAD(Y_MINUS,Y_0,Y_PLUS: REAL;
               VAR XE,YE,ZERO1,ZERO2: REAL; VAR NZ: INTEGER);

  VAR A,B,C,DIS,DX: REAL;

BEGIN
  NZ := 0;
  A := 0.5*(Y_MINUS+Y_PLUS)-Y_0; B := 0.5*(Y_PLUS-Y_MINUS); C := Y_0;
  XE := -B/(2.0*A); YE := (A*XE + B) * XE + C;
  DIS := B*B - 4.0*A*C; (* discriminant of y = axx+bx+c *)
  IF (DIS >= 0) THEN (* parabola intersects x-axis *)
    BEGIN
      DX := 0.5*SQRT(DIS)/ABS(A); ZERO1 := XE-DX; ZERO2 := XE+DX;
      IF (ABS(ZERO1) <= +1.0) THEN NZ := NZ + 1;
      IF (ABS(ZERO2) <= +1.0) THEN NZ := NZ + 1;
      IF (ZERO1 < -1.0) THEN ZERO1:=ZERO2;
    END;
  END;
END;
```

3.8 The SUNSET Program

The program SUNSET calculates the rising and setting times of the Sun and the Moon, as well as the beginning and ending of nautical twilight over a period of 10 days. Data to be entered are the starting date, the geographical

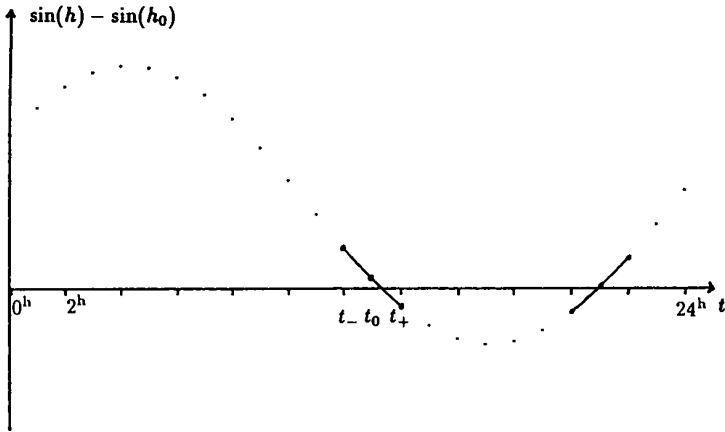


Fig. 3.4. Determining rising and setting times

coordinates of the observing site, and the difference between Local Time and Universal Time.

The search for the times of the various events is made by the method shown diagrammatically in Fig. 3.4. The `SIN_ALT` procedure calculates the sine of the solar or lunar altitude at hourly intervals. These values are interpolated in `QUAD` and examined for zero points. If a zero point is found, and the Sun or Moon was below the horizon at the beginning of the day, then the event is a sunrise or moonrise. Otherwise, the Sun or Moon is setting. Finally, it may happen that two zero points are discovered in the interval being examined. In order to determine which of these two points is a rising and which a setting, we check to see if the altitude at the vertex is above or below the horizon. The process continues until the end of the day. It is then possible to determine whether the celestial body being considered is circumpolar, or whether it remains below the horizon for the entire day.

The times of nautical twilight are calculated in a similar manner, in that a constant value of $\sin(-12^\circ)$ is subtracted from the sine of the height of the horizon. If the times of civil or astronomical twilight are required, then the value of `SINHO[3]` may be appropriately altered.

```
(*-----*)
(*                               SUNSET                               *)
(*      solar and lunar rising and setting times                      *)
(*                               version 93/07/01                       *)
(*-----*)
```

```
PROGRAM SUNSET(INPUT,OUTPUT);
```

```
VAR  ABOVE,RISE,SETT           : BOOLEAN;
      DAY,MONTH,YEAR, I,IOBJ,NZ : INTEGER;
      LAMBDA,ZONE,PHI,SPHI,CPhi : REAL;
```

```

TSTART,DATE,HOUR,HH,UTRISE,UTSET      : REAL;
Y_MINUS,Y_0,Y_PLUS,ZERO1,ZERO2,XE,YE  : REAL;
SINH0                                  : ARRAY[1..3] OF REAL;

```

```

(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS, QUAD *)
(* MJD, CALDAT, LMST *)
(* MINISUN, MINIMOON *)
(*-----*)

```

```

(*-----*)
(* SIN_ALT: sin(altitude) *)
(* IOBJ: 1=moon, 2=sun *)
(*-----*)

```

```

FUNCTION SIN_ALT(IOBJ:INTEGER;MJDO,HOUR,LAMBDA,CPhi,SPHI:REAL): REAL;
VAR MJD,T,RA,DEC,TAU: REAL;
BEGIN
  MJD := MJDO + HOUR/24.0;
  T := (MJD-51544.5)/36525.0;
  IF (IOBJ=1)
    THEN MINI_MOON(T,RA,DEC)
    ELSE MINI_SUN (T,RA,DEC);
  TAU := 15.0 * (LMST(MJD,LAMBDA) - RA);
  SIN_ALT := SPHI*SN(DEC) + CPhi*CS(DEC)*CS(TAU);
END;

```

```

(*-----*)
(* WHM: write time in hours and minutes *)
(*-----*)

```

```

PROCEDURE WHM(UT:REAL);
VAR H,M:INTEGER;
BEGIN
  UT := TRUNC(UT*60.0+0.5)/60.0; (* round to 1 min *)
  H:=TRUNC(UT); M:=TRUNC(60.0*(UT-H)+0.5); WRITE(H:5,' ',M:2,' ');
END;

```

```

(*-----*)
(* GETINP: read input data *)
(*-----*)

```

```

PROCEDURE GETINP(VAR DATE,LAMBDA,PHI,ZONE: REAL);
VAR D,M,Y: INTEGER;
BEGIN
  WRITELN;
  WRITELN('          SUNSET: solar and lunar rising and setting times ');
  WRITELN('                      version 93/07/01                      ');
  WRITELN('          (c) 1993 Thomas Pfleger, Oliver Montenbruck          ');
  WRITELN;
  WRITELN;
  WRITE (' First date (yyyy mm dd)                                ... '); READLN(Y,M,D);
  WRITELN;
  WRITE (' Observing site: longitude (1<0 east) ... '); READLN(LAMBDA);
  WRITE ('                      latitude ... '); READLN(PHI);
  WRITE ('                      local time - UT (h) ... '); READLN(ZONE);
  WRITELN;

```

```

WRITELN; WRITE('    Date');
WRITELN('        Moon                Sun                Twilight ');
WRITELN; WRITE('        ');
WRITELN('        rise/set                rise/set                end/beginning');
WRITELN;
ZONE := ZONE /24.0;
DATE := MJD(D,M,Y,0)-ZONE;
END;
(*-----*)

BEGIN (* SUNSET *)

SINHO[1] := SN ( +8.0/60.0); (* moonrise                at h= +8'        *)
SINHO[2] := SN (-50.0/60.0); (* sunrise                at h=-50'        *)
SINHO[3] := SN ( -12.0 ); (* nautical twilight at h=-12 degrees *)

GETINP (TSTART,LAMBDA,PHI,ZONE);

SPHI := SN(PHI);  CPHI := CS(PHI);

FOR I:=0 TO 9 DO  (* loop over 10 subsequent days *)

    BEGIN

        DATE := TSTART + I;
        CALDAT(DATE+ZONE,DAY,MONTH,YEAR,HH);
        WRITE(YEAR:5,'/',MONTH:2,'/',DAY:2,' '); (* print current date *)

        FOR IOBJ := 1 TO 3 DO

            BEGIN

                HOUR := 1.0;
                Y_MINUS := SIN_ALT(IOBJ,DATE,HOUR-1.0,LAMBDA,CPHI,SPHI)
                    - SINHO[IOBJ];

                ABOVE := (Y_MINUS>0.0); RISE := FALSE; SETT := FALSE;

                (* loop over search intervals from [0h-2h] to [22h-24h] *)
                REPEAT

                    Y_0      := SIN_ALT(IOBJ,DATE,HOUR,LAMBDA,CPHI,SPHI)
                        - SINHO[IOBJ];
                    Y_PLUS := SIN_ALT(IOBJ,DATE,HOUR+1.0,LAMBDA,CPHI,SPHI)
                        - SINHO[IOBJ];

                    (* find parabola through three values Y_MINUS,Y_0,Y_PLUS *)
                    QUAD ( Y_MINUS,Y_0,Y_PLUS, XE,YE, ZERO1,ZERO2, NZ );

                    CASE (NZ) OF
                        0: ;
                        1: IF (Y_MINUS<0.0)
                            THEN BEGIN UTRISE:=HOUR+ZERO1;  RISE:=TRUE; END

```

```

        ELSE BEGIN UTSET :=HOURL+ZERO1;  SETT:=TRUE; END;
2: BEGIN
    IF (YE<0.0)
        THEN BEGIN UTRISE:=HOURL+ZERO2;UTSET:=HOURL+ZERO1; END
        ELSE BEGIN UTRISE:=HOURL+ZERO1;UTSET:=HOURL+ZERO2; END;
        RISE:=TRUE; SETT:=TRUE;
    END;
END;
Y_MINUS := Y_PLUS;      (* prepare for next interval *)
HOURL := HOURL + 2.0;

UNTIL ( (HOURL=25.0) OR (RISE AND SETT) );

IF (RISE OR SETT) (* output *)
THEN
    BEGIN
        IF RISE THEN WHM(UTRISE) ELSE WRITE('----- ':9);
        IF SETT THEN WHM(UTSET)  ELSE WRITE('----- ':9);
    END
ELSE
    BEGIN
        IF ABOVE
            THEN CASE IOBJ OF
                1,2: WRITE ('    always visible ');
                3:  WRITE ('    always bright ');
            END
        ELSE CASE IOBJ OF
                1,2: WRITE ('    always invisible');
                3:  WRITE ('    always dark ');
            END;
        END;
    END;

END;

WRITELN;

END; (* end of loop over 10 days *)

WRITELN; WRITE (' all times in local time ( = UT ');
IF ZONE>=0 THEN WRITE('+'); WRITELN(ZONE*24.0:5:1,'h '));

END.

```

(*-----*)

We will demonstrate the use of SUNSET with some examples, which will show some of the factors that arise in calculating rising and setting times.

SUNSET calculates the desired times for a period of ten days, beginning with the date entered. The geographical longitude is entered with negative values for those places east of the Greenwich meridian. Finally the difference between Local Time and UT is entered in hours. A value of 1, for example,

would be used for Central European Time (CET), or 2 for Central European Summer Time (CEST). For time zones in North America, negative values are entered: -5 for Eastern Standard Time (EST), for example, or -8 for Pacific Standard Time (PST).

Let us, for example, calculate the rising and setting times of the Sun and Moon for Munich ($\lambda = 11^{\circ}6'$ East, $\varphi = 48^{\circ}1'$ North), beginning with 1989 March 23. We require the output for Central European Time (1^h different from Universal Time). First the starting date must be entered in the form: Year, Month, Day. Then SUNSET asks for the geographical coordinates and the difference between LT and UT. (In the following, all data entered are shown in *italic*.) From the data that we have entered, SUNSET determines the rising and setting times of the Sun and Moon, as well as the beginning and end of twilight. As the value for the altitude of the Sun that is used in the program is set at -12° , the output here and in the following example is for *nautical* twilight.

SUNSET: solar and lunar rising and setting times
version 93/07/01
(c) 1993 Thomas Pflieger, Oliver Montenbruck

First date (yyyy mm dd) ... 1989 3 23

Observing site: longitude (1<0 east) ... -11.6
latitude ... 48.1
local time - UT (h) ... 1.0

Date	Moon		Sun		Twilight	
	rise/set		rise/set		end/beginning	
1989/ 3/23	19:57	6:13	6:11	18:30	5: 3	19:39
1989/ 3/24	21: 5	6:28	6: 9	18:32	5: 1	19:40
1989/ 3/25	22:15	6:45	6: 7	18:33	4:59	19:42
1989/ 3/26	23:26	7: 6	6: 5	18:35	4:56	19:44
1989/ 3/27	-----	7:33	6: 2	18:36	4:54	19:45
1989/ 3/28	0:34	8: 9	6: 0	18:38	4:52	19:47
1989/ 3/29	1:38	8:58	5:58	18:39	4:50	19:48
1989/ 3/30	2:31	10: 0	5:56	18:41	4:48	19:50
1989/ 3/31	3:14	11:14	5:54	18:42	4:45	19:52
1989/ 4/ 1	3:47	12:35	5:52	18:43	4:43	19:53

all times in local time (= UT + 1.0 h)

It will be seen that there is no entry in the column under 'Moon rise' for 1989 March 27. On that night the Moon rose after midnight, so it is recorded under March 28. In general, there is one day per month on which it neither rises or sets.

In order to illustrate another interesting effect, let us calculate the rising and setting times for a fictitious point in Europe at latitude 65° , choosing 1989 June 15 as the starting date and Central European Summer Time as the local time. SUNSET gives the following output:

SUNSET: solar and lunar rising and setting times
version 93/07/01
(c) 1993 Thomas Pflieger, Oliver Montenbruck

First date (yyyy mm dd) ... 1989 6 15

Observing site: longitude (1<0 east) ... -10.0
latitude ... 65.0
local time - UT (h) ... 2.0

Date	Moon		Sun		Twilight
	rise/set		rise/set		end/beginning
1989/ 6/15	19:58	1: 0	2:24	0:16	always bright
1989/ 6/16	22:26	23:53	2:23	0:18	always bright
1989/ 6/17	always invisible		2:22	0:19	always bright
1989/ 6/18	always invisible		2:21	0:20	always bright
1989/ 6/19	always invisible		2:20	0:21	always bright
1989/ 6/20	always invisible		2:20	0:22	always bright
1989/ 6/21	2:39	3:24	2:20	0:23	always bright
1989/ 6/22	1:35	6:21	2:20	0:23	always bright
1989/ 6/23	1:15	8:29	2:21	0:23	always bright
1989/ 6/24	1: 1	10:25	2:22	0:22	always bright

all times in local time (= UT + 2.0 h)

In this case, after setting on June 16 at $23^{\text{h}}53^{\text{m}}$, the Moon remains below the horizon and is therefore invisible until June 21 at $2^{\text{h}}39^{\text{m}}$. Correspondingly, at high latitudes the Moon may remain above the horizon for days.

Naturally, similar events can occur with the Sun, which we call the polar night or polar daylight ('midnight Sun'). Using SUNSET we can estimate how long polar night or polar daylight last at a specific place. In order to do this we vary the period over which calculations are carried out, until the Sun is first shown as 'always invisible'. Polar night begins on the corresponding day. The end of the polar night can be found in a precisely analogous way.

The result 'always bright' in the 'Twilight' column indicates that it never becomes fully dark. This can even occur at a latitude of only 50° , when we determine astronomical instead of nautical or civil twilight.

3.9 The PLANRISE Program

In the previous sections we became acquainted with the necessary tools for computing rising and setting times by the method of quadratic interpolation. The latter may also be used to handle exceptional cases of rising and setting time calculations, which may otherwise present severe difficulties. Nevertheless, we also want to describe a program that illustrates the iterative method of obtaining planetary rising and setting times. The commented source code of PLANRISE is provided on the enclosed disk.

The function of PLANRISE is based on the equations given in Sect. 3.6. For the computation of planetary positions we make use of the POSITION routine, which is described in Sect. 5.2. To compute the rising and setting times of a planet we first determine its (geocentric) equatorial coordinates α and δ for an arbitrary time on the given day. Equation (3.11) then yields the value $\cos \tau(\varphi, \delta)$ for the hour angle at the instant of rising or setting. If the right-hand side results in a value whose magnitude is larger than one, the planet never rises or sets on that day. From the declination and the geographical latitude, PLANRISE then determines whether the planet is continuously above or below the horizon.

Because the horizontal parallax and the apparent diameter have virtually no impact on the rising and setting times, a value of $h_{R/S} = -0^\circ 34'$ for the geometric altitude at the time of rising or setting is employed in the computation, which thus takes only refraction into account. The iterative equation for determining the rising and setting times is then given by

$$t_{i+1} = t_i - 0.9973(\Theta(t_i) - \alpha(t_i) \pm \tau(\varphi, \delta(t_i))) \quad (3.16)$$

where

$$\tau = \arccos \frac{\sin h_{A/U} - \sin \varphi \sin \delta}{\cos \varphi \cos \delta} \quad (3.17)$$

Here the positive sign applies for the computation of rising times. $\Theta(t_i)$ and $\alpha(t_i)$ denote the local sidereal time and the planet's right ascension at the time t_i of the i th approximation. The conversion factor 0.9973 is used to convert between universal time and sidereal time, as mentioned earlier.

As with the computation of rising and setting times, we may also employ the relationship

$$t_{i+1} = t_i - 0.9973(\Theta(t_i) - \alpha(t_i)) \quad (3.18)$$

to determine the time of culmination of a planet.

In order to reduce computing times, various simplifications are used inside PLANRISE. Linear interpolation, for example, is used to obtain the planetary coordinates α and δ , as well as the local sidereal time Θ , from precomputed values at 0^h and 24^h on the same day.

As an example we will compute the rising and setting times of the planets on 1994 January 1 in Munich ($\lambda = 11.6^\circ$ East, $\varphi = 48.1^\circ$ North). All input data for the PLANRISE program are shown in *italic*.

PLANRISE: planetary and solar rising and setting times

version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Date (yyyy mm dd) ... 1994 1 1

Observing site: longitude (1<0 east) ... *-11.6*

latitude ... *48.1*

local time - UT (h) ... *1.0*

	rise	culmination	set
Mercury	08:11	12:12	16:13
Venus	07:53	12:01	16:09
Sun	08:04	12:17	16:30
Mars	08:06	12:11	16:17
Jupiter	03:02	08:02	13:02
Saturn	10:29	15:29	20:28
Uranus	08:48	13:04	17:19
Neptune	08:38	12:58	17:19
Pluto	03:47	09:22	14:58

all times in local time (= UT + 1.0h)

4. Cometary Orbits

Whereas the orbits of the major planets are often calculated and published years in advance, predictions of cometary orbits are generally restricted to short-term ephemerides and the determination of orbital elements. The five or six orbital elements describe the orbit in space and time in a significantly shorter form than a long listing of positions. In general, the orbital elements of newly discovered comets are published by the International Astronomical Union (IAU) in the *IAU Circulars*. In addition, there are various catalogues that give the corresponding data for periodic comets and minor planets.

From the orbital elements, and using a suitable program, it is possible to calculate the position of a comet relative to the Earth and the Sun at any time. The COMET program that is described in this chapter is not restricted to the calculation of elliptical orbits, such as those of the planets and minor planets. It is also suitable for calculating cometary orbits that are parabolic, near-parabolic, or hyperbolic.

The following sections discuss the motion of comets relative to the Sun and to the ecliptic, neglecting perturbations by the major planets. Taking them into account would require an enormous, additional amount of work. The various coordinate conversions that are required for conversion into geocentric, equatorial coordinates have already been described in Chap. 2. Details of the following procedures, which are used by COMET, may be found there:

SN, CS, ATN, ATN2, CART, POLAR, DMS, CUBR,
MJD, CALDAT
ECLEQU, PMATECL, PRECART,
SUN200.

4.1 Form and Orientation of the Orbit

The motion of comets around the Sun is governed by the same laws that control the orbits of the major planets:

1. The orbit is an ellipse, a parabola, or a hyperbola (i.e., a conic section), with the Sun at the focus (or at one of the two foci in the case of an ellipse). This means, in particular, that the orbit lies in a fixed plane.
2. The line joining the Sun and the comet (the radius vector) sweeps out equal areas in equal periods of time. This is known as the *law of equal areas*.

These laws were discovered by Johannes Kepler, and were later explained by Newton's theory of universal gravitation. Nevertheless, Kepler's Laws are valid only when the influence of the major planets on comets can be neglected in comparison with the Sun's gravity.

The orbit lies in a single plane because of the fact that the Sun's gravitational attraction always acts along the line joining the Sun and the comet (this is known as a centripetal force). In a short interval Δt , the comet moves at a velocity $\dot{r}$ over a distance $\dot{r} \cdot \Delta t$ between point r and point $r + \dot{r} \cdot \Delta t$ (Fig. 4.1). Taken together with the Sun, these two points determine a plane in space. Because the force of the Sun only acts in this plane, however, subsequent motion of the comet continues to lie in this plane. The law of equal areas is also a direct consequence of centripetal force. Proof of this requires a moderate amount of mathematics, so it will not be shown here.

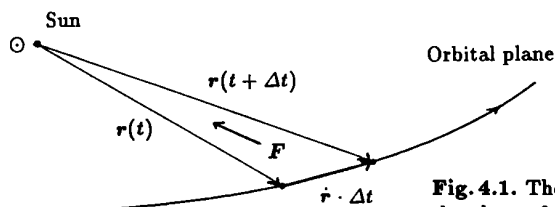


Fig. 4.1. The force on comets does not alter the plane of their orbits

The exact form of the orbit is determined by the decrease in the gravitational force with the square of the distance ($F \sim 1/r^2$). The three basic forms—the ellipse, parabola, and hyperbola—are shown in Fig. 4.2. While comets in elliptical orbits are periodic, making repeated returns, those in parabolic or hyperbolic orbits are close to the Sun only for a single, short period of time. The closest point on the orbit to the Sun is known as *perihelion*. The line joining perihelion to the Sun (known as the *line of apsides*) divides the orbit into two symmetrical halves.

A point on the orbit is defined by the distance from the Sun r , and the true anomaly¹ ν . The latter is the angle between the Sun-comet line and the Sun-perihelion direction. The relationship between r and ν is given by the equation of the conic section :

$$r = \frac{p}{1 + e \cdot \cos(\nu)} \quad (4.1)$$

The value p , which is the distance for $\nu = 90^\circ$, is the semi-latus rectum (sometimes known as the *orbital parameter*) and it defines the size of the orbit. The *eccentricity* e , indicates how greatly the orbit deviates from a circle ($e = 0$). The higher the value, the more elongated the orbit, which finally becomes a parabola ($e = 1$), or a hyperbola ($e > 1$).

¹In celestial mechanics, 'anomaly' is used to mean an angle. In addition to the true anomaly, other parameters known as the mean anomaly and the eccentric anomaly are used.

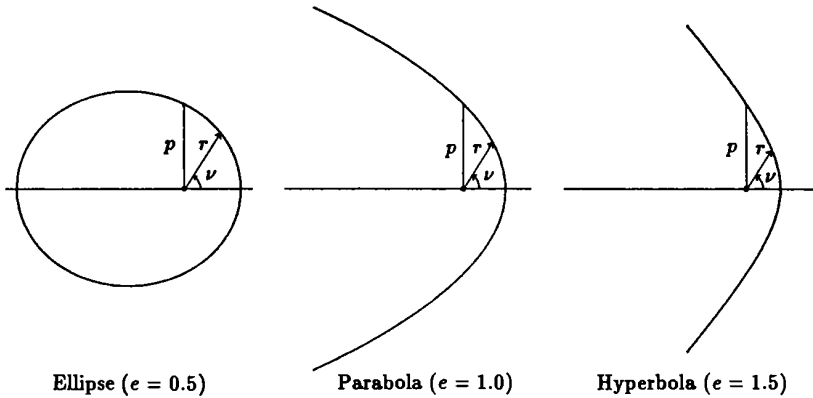


Fig. 4.2. Conic sections with eccentricities $e = 0.5$, $e = 1.0$ and $e = 1.5$ with the same orbital parameter p

A rather more useful value than the orbital parameter is the semi-major axis, which is, however, only defined when $e \neq 1$:

$$a = \frac{p}{1 - e^2} \quad , \quad p = a(1 - e^2) \quad . \quad (4.2)$$

For an elliptical orbit, a obviously equals the average of the least and greatest distances from the Sun:

$$r_{\min} + r_{\max} = \frac{p}{1 + e} + \frac{p}{1 - e} = \frac{2p}{1 - e^2} = 2a \quad .$$

It should be noted that the semi-major axis of a hyperbola—where $e > 1$ by definition—is negative. For parabolic orbits ($e \approx 1$), the perihelion distance $q = r_{\min}$ is used instead of a :

$$q = \frac{p}{1 + e} \quad . \quad (4.3)$$

In an orbit that is exactly a parabola, $q = p/2$.

4.2 Position in the Orbit

Although the equation for the conic section gives every possible position at which a comet may be found during its orbit, it does not establish at what time this will occur. In order to solve this problem, we use the law of equal areas. Using a closed elliptical orbit as an example, we will now discuss the major steps in this calculation.

An ellipse of eccentricity e may be obtained by reducing a single diameter of a circle by a factor of $\sqrt{1 - e^2}$. The total area S of the ellipse with semi-major axis a is therefore smaller than that of a circle of radius a by this same factor, so we have:

$$S = (\pi a^2)\sqrt{1 - e^2} \quad .$$

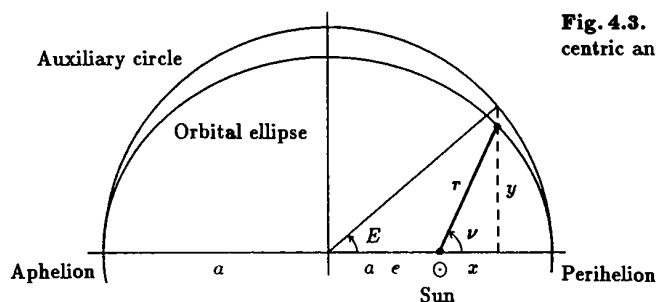


Fig. 4.3. The way in which eccentric anomaly E is defined

The line joining the Sun and the comet (the radius vector) sweeps out this area in the orbital period T , and the sector

$$S(t) = (\pi a^2) \sqrt{1 - e^2} \left(\frac{t - t_0}{T} \right) \quad (4.4)$$

in the time t after perihelion passage t_0 . If it is now possible to describe the area of this sector S in terms of the position in the orbit (r, ν) , then we have established the required connection between the position in the orbit and time t . For a circular orbit this is extremely simple, because S is directly proportional to the angle ν and to the square of the semi-diameter. For an ellipse, however, we need to introduce an auxiliary parameter—the *eccentric anomaly*. Its geometrical significance can be seen from Fig. 4.3. If the x -axis is chosen as shown there, a point P on the ellipse has an x -coordinate of $x = a \cdot \cos E - a \cdot e$. The $a \cdot e$ term indicates the distance between the centre of the ellipse and the focus at which the Sun is situated. The y -coordinate is less than that of the point P' on the circumscribed circle by the compression factor $\sqrt{1 - e^2}$, so $y = \sqrt{1 - e^2} \cdot a \cdot \sin E$. We then have

$$\begin{aligned} x &= r \cos \nu = a (\cos E - e) \\ y &= r \sin \nu = a \sqrt{1 - e^2} \sin E \end{aligned} \quad (4.5)$$

In order to calculate S , two auxiliary areas may be considered, and these are illustrated in greater detail in Fig. 4.4:

$$\text{Sector of the circle } (P'O\Pi) : \quad S_1 = \pi a^2 E / 360^\circ$$

$$\text{Sector of the Ellipse } (PO\Pi) : \quad S_2 = \sqrt{1 - e^2} \cdot S_1$$

$$\text{Triangle } (O \odot P) : \quad S_3 = \frac{1}{2} y \cdot (a \cdot e) = \frac{1}{2} a^2 e \sqrt{1 - e^2} \sin E$$

$$\text{Sector } (P \odot \Pi) : \quad S = S_2 - S_3$$

By using the eccentric anomaly, the area S can be expressed in the form

$$S(t) = \frac{1}{2} a^2 \sqrt{1 - e^2} \left(E \frac{\pi}{180^\circ} - e \sin E \right) \quad (4.6)$$

If this is compared with the formula for $S(t)$ given above, then from the relationship between the eccentric anomaly and the time, we obtain *Kepler's Equation*:

$$E(t) - \frac{180^\circ}{\pi} e \sin E(t) = M(t) \quad (4.7)$$

where

$$M(t) = 360^\circ \frac{(t - t_0)}{T} \quad (4.8)$$

In order to determine the position of the comet in its orbit at any given instant, we must solve this equation for E , and substitute the value thus obtained into Equation (4.5) to derive x and y . This process will be described in detail in the section about the mathematical treatment of Kepler's Equation.

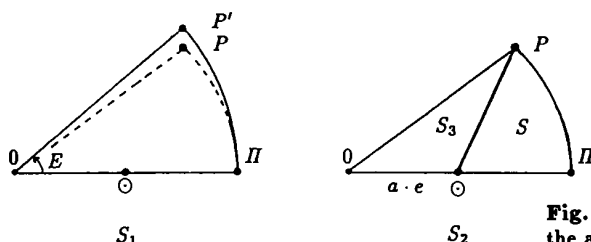


Fig. 4.4. The method of obtaining the area S of the sector

Until now, the orbital period has been treated as an independent variable. In fact it depends on the semi-major axis as given by *Kepler's Third Law*:

$$\frac{a^3}{T^2} = \frac{GM_\odot}{4\pi^2} \quad \text{where} \quad GM_\odot = 1.32712 \cdot 10^{20} \text{ m}^3 \text{ s}^{-2} \quad (4.9)$$

The constant on the right-hand side of this equation contains the product $GM_\odot$, the gravitational constant and one solar mass. The units in which the value is given here—the metre (m) and second (s)—are rarely used in practice. The usual units are the day (1 d = 86400 s) and the astronomical unit (1 AU $\approx$ 149.59787 million km). The AU originally represented the semi-major axis of the Earth's orbit. Later the IAU decided that the astronomical unit should be defined such that the product of the gravitational constant and the solar mass should have a fixed value

$$\begin{aligned} GM_\odot &= k^2 \text{ AU}^3 \text{ d}^{-2} \quad \text{where} \\ k &= 0.01720209895 \\ k^2 &\approx 2.959122083 \cdot 10^{-4} \end{aligned}$$

k is known as the Gaussian gravitational constant. Although at first sight this definition appears somewhat unusual, it means that one AU is defined

irrespective of any possible long-term changes in the diameter of the Earth's orbit.

As well as the position of the comet in its orbit, we shall also require its velocity. The equations for determining this may be given here:

$$\begin{aligned}\dot{x} &= -\sqrt{\frac{GM_{\odot}}{a}} \frac{\sin E}{1 - e \cos E} \\ \dot{y} &= +\sqrt{\frac{GM_{\odot}(1 - e^2)}{a}} \frac{\cos E}{1 - e \cos E} .\end{aligned}\quad (4.10)$$

For hyperbolic orbits we can determine equations similar to those for an elliptical orbit. Instead of the angular functions, we use the hyperbolic functions $\sinh x = (e^x - e^{-x})/2$ and $\cosh x = (e^x + e^{-x})/2$. We therefore have:

$$\begin{aligned}x = r \cos \nu &= |a| (e - \cosh H) \\ y = r \sin \nu &= |a| \sqrt{e^2 - 1} \sinh H\end{aligned}\quad (4.11)$$

and

$$\begin{aligned}\dot{x} &= -\sqrt{\frac{GM_{\odot}}{|a|}} \frac{\sinh H}{e \cosh H - 1} \\ \dot{y} &= +\sqrt{\frac{GM_{\odot}(e^2 - 1)}{|a|}} \frac{\cosh H}{e \cosh H - 1} .\end{aligned}\quad (4.12)$$

The parameter H , which corresponds to the eccentric anomaly, allows us to obtain a modified form of Kepler's Equation:

$$e \sinh H - H = M_h = \sqrt{\frac{GM_{\odot}}{|a|^3}} \cdot (t - t_0) .\quad (4.13)$$

With parabolic orbits, x and y are first expressed in terms of $\tan(\nu/2)$:

$$\begin{aligned}x = r \cos \nu &= q \cdot \left(1 - \tan^2 \left(\frac{\nu}{2}\right)\right) \\ y = r \sin \nu &= 2q \cdot \tan \left(\frac{\nu}{2}\right) .\end{aligned}\quad (4.14)$$

As mentioned earlier, q is the perihelion distance. The true anomaly ν is expressed in terms of time by *Barker's Equation*:

$$\tan \left(\frac{\nu}{2}\right) + \frac{1}{3} \tan^3 \left(\frac{\nu}{2}\right) = \sqrt{\frac{GM_{\odot}}{2q^3}} (t - t_0) .\quad (4.15)$$

This is a third-order equation of $\tan(\nu/2)$. In contrast to Kepler's Equation it can be solved directly for ν for a given time t . If we set

$$A = \frac{3}{2} \sqrt{\frac{GM_{\odot}}{2q^3}} (t - t_0) \quad \text{and} \quad B = \sqrt[3]{A + \sqrt{A^2 + 1}} ,$$

then

$$\tan(\nu/2) = B - 1/B , \quad \nu = 2 \arctan(B - 1/B) .$$

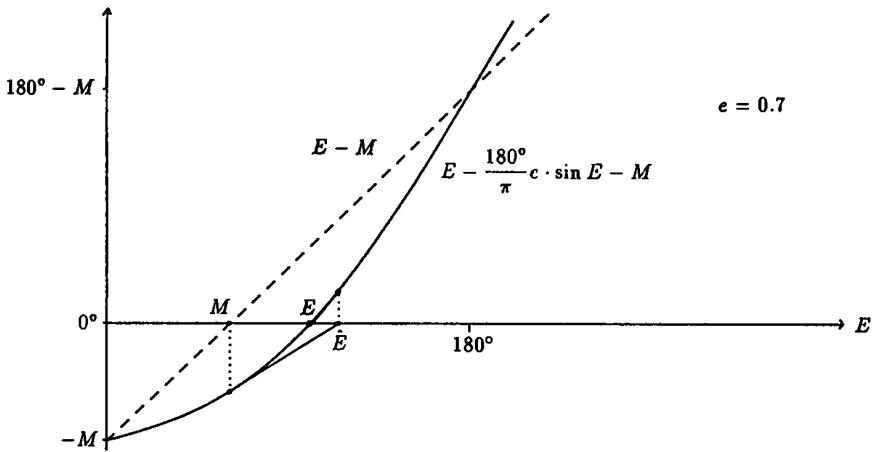


Fig. 4.5. Solving Kepler's Equation by Newton's method

4.3 Mathematical Treatment of Kepler's Equation

Kepler's Equation cannot be solved analytically with respect to time, using the eccentric anomaly as a function of the mean anomaly. There are, however, appropriate mathematical procedures that enable this to be done.

We first calculate the mean anomaly M at a given time t . For low eccentricities this differs only slightly from the eccentric anomaly E . It can therefore be used as a first approximation E_0 for a subsequent iteration. If Kepler's Equation is slightly rearranged, we can see that in order to solve it, the root of the function

$$f(E) = E - \frac{180^\circ}{\pi} e \sin E - M(t)$$

has to be determined. One method of finding the point at which a function $f(x)$ is zero, is *Newton's method* (see Fig. 4.5). The tangent at a point x close to the zero point, and with a slope of $f'(x)$ is used as an approximation, and its intersection with the x -axis is calculated from:

$$\hat{x} = x - \frac{f(x)}{f'(x)}.$$

In general one obtains an improved approximation $\hat{x}$ to the zero point of the function f . In the case of Kepler's Equation, the iterative expression has the form

$$\hat{E} = E - \frac{E - (180^\circ/\pi) \cdot e \cdot \sin E - M(t)}{1 - e \cdot \cos E}.$$

Here E must be given in degrees. If one is working in radians, then the factor $180^\circ/\pi$ must be replaced by 1. For orbits that are close to a circle,

three steps are typically sufficient to obtain a solution for E accurate to ten decimal places. Nevertheless, one should be careful not to apply Newton's method indiscriminately in cases of high eccentricity. It is suggested that readers should see for themselves what happens when the iteration is carried out with $e = 0.99$ and $E_0 = M = 5^\circ$. Safety suggests that here another initial value should be used, namely $E_0 = \pi = 180^\circ$. This initial value is used in the ECCANOM function for eccentricities above $e = 0.8$.

```
(*-----*)
(* ECCANOM: calculation of the eccentric anomaly E=ECCANOM(MAN,ECC) *)
(*      from the mean anomaly MAN and the eccentricity ECC.      *)
(*      (solution of Kepler's equation by Newton's method)      *)
(*      (E, MAN in degrees)                                      *)
(*-----*)
FUNCTION ECCANOM(MAN,ECC:REAL):REAL;
  CONST PI=3.141592654; TWOPI=6.283185308; RAD=0.0174532925199433;
        EPS = 1E-11; MAXIT = 15;
  VAR M,E,F: REAL; I: INTEGER;
  BEGIN
    M:=MAN/360.0; M:=TWOPI*(M-TRUNC(M)); IF M<0 THEN M:=M+TWOPI;
    IF (ECC<0.8) THEN E:=M ELSE E:=PI;
    F := E - ECC*SIN(E) - M; I:=0;
    WHILE ( ABS(F)>EPS) AND (I<MAXIT) ) DO
      BEGIN
        E := E - F / (1.0-ECC*COS(E)); F := E-ECC*SIN(E)-M; I:=I+1;
      END;
    ECCANOM:=E/RAD;
    IF (I=MAXIT) THEN WRITELN(' convergence problems in ECCANOM');
  END;
(*-----*)
```

It is now possible to determine the eccentric anomaly by calling this function just once. The remaining steps that are required to calculate an orbital position on the ellipse are included in the ELLIP procedure:

```
(*-----*)
(* ELLIP: calculation of the position and velocity vectors      *)
(*      for elliptical orbits                                    *)
(*-----*)
(*      M      mean anomaly in degrees      X,Y      position   (in AU) *)
(*      A      semimajor axis                VX,VY    velocity  (in AU/day) *)
(*      ECC    eccentricity                  *)
(*-----*)
PROCEDURE ELLIP(M,A,ECC:REAL;VAR X,Y,VX,VY:REAL);
  CONST KGAUSS = 0.01720209895;
  VAR K,E,C,S,FAC,RHO: REAL;
  BEGIN
    K := KGAUSS/SQRT(A); E := ECCANOM(M,ECC); C:=COS(E); S:=SIN(E);
    FAC:= SQRT((1.0-ECC)*(1+ECC)); RHO:=1.0-ECC*C;
    X := A*(C-ECC); Y := A*FAC*S; VX:=-K*S/RHO; VY:=K*FAC*C/RHO;
  END;
(*-----*)
```

Kepler's Equation can be solved in a similar manner for hyperbolic orbits. Here the iterative expression is:

$$\dot{H} = H - \frac{e \cdot \sinh H - H - M_h(t)}{e \cdot \cosh H - 1}.$$

A suitable initial value is

$$H_0 = \begin{cases} +\ln(+1.8 + 2M_h/e) & \text{for } M_h \geq 0 \\ -\ln(+1.8 - 2M_h/e) & \text{for } M_h < 0 \end{cases}.$$

```
(*-----*)
(* HYPANOM: calculation of the eccentric anomaly H=HYPANOM(MH,ECC) from *)
(*      mean anomaly MH and eccentricity ECC for *)
(*      hyperbolic orbits *)
(*-----*)
FUNCTION HYPANOM(MH,ECC:REAL):REAL;
  CONST EPS=1E-10; MAXIT=15;
  VAR  H,F,EXH,SINHH,COSHH: REAL;
       I : INTEGER;
  BEGIN
    H:=LN(2.0*ABS(MH)/ECC+1.8);
    IF (MH<0.0) THEN H:=-H;
    EXH:=EXP(H);
    SINHH:=0.5*(EXH-1.0/EXH); COSHH:=0.5*(EXH+1.0/EXH);
    F := ECC*SINHH-H-MH; I:=0;
    WHILE ( (ABS(F)>EPS*(1.0+ABS(H+MH))) AND (I<MAXIT) ) DO
      BEGIN
        H := H - F / (ECC*COSHH-1.0);
        EXH:=EXP(H); SINHH:=0.5*(EXH-1.0/EXH); COSHH:=0.5*(EXH+1.0/EXH);
        F := ECC*SINHH-H-MH; I:=I+1;
      END;
    HYPANOM:=H;
    IF (I=MAXIT) THEN WRITELN(' convergence problems in HYPANOM');
  END;
(*-----*)
```

The rectangular coordinates of the comet along its orbit can be calculated with the HYPERB procedure from the formulae given in the previous section. Unlike ELLIP, the times of perihelion passage and the required instant are entered, instead of the mean anomaly M_h . The reason for the different choice of variables passed is that the ELLIP procedure also has to be suitable for the calculation of the orbits of minor planets and planets. With these bodies, unlike the situation with comets, the time of perihelion passage is rarely employed as an orbital element. Instead, the mean anomaly M_{Ep} and the mean daily motion n at a specific epoch t_{Ep} are generally available. The mean anomaly $M = M_{Ep} + n(t - t_{Ep})$ at a given time t can then be used directly as input to ELLIP. On the other hand, hyperbolic orbits are generally encountered only among cometary orbits, where it is customary for the time of perihelion passage to be given as one of the orbital elements.

```

(*-----*)
(* HYPERB: calculation of the position and velocity vectors *)
(*      for hyperbolic orbits *)
(*      *)
(* TO   time of perihelion passage          X,Y   position *)
(* T    time                               VX,VY  velocity *)
(* A    semimajor axis (arbitrary sign) *)
(* ECC  eccentricity *)
(* (TO,T in julian centuries since J2000) *)
(*-----*)
PROCEDURE HYPERB(TO,T,A,ECC:REAL;VAR X,Y,VX,VY:REAL);
  CONST KGAUSS = 0.01720209895;
  VAR K,MH,H,EXH,COSHH,SINHH,RHO,FAC: REAL;
  BEGIN
    A := ABS(A);
    K := KGAUSS / SQRT(A);
    MH := K*36525.0*(T-TO)/A;
    H := HYPANOM(MH,ECC);
    EXH := EXP(H);  COSHH:=0.5*(EXH+1.0/EXH);  SINHH:=0.5*(EXH-1.0/EXH);
    FAC := SQRT((ECC+1.0)*(ECC-1.0));  RHO := ECC*COSHH-1.0;
    X := A*(ECC-COSHH);  Y := A*FAC*SINHH;
    VX := -K*SINHH/RHO;  VY := K*FAC*COSHH/RHO;
  END;
(*-----*)

```

4.4 Near-Parabolic Orbits

As we indicated in the previous section, the calculation of elliptical or hyperbolic orbits with eccentricities close to one, and with small anomalies, is not always easy. We can avoid these difficulties from the start by making use of the fact that orbits with the characteristics described are very similar to a parabola. The method used here is that devised by Karl Stumpff.

From Kepler's equation

$$E(t) - e \sin E(t) = \sqrt{\frac{GM_{\odot}}{a^3}} \cdot (t - t_0) \quad ,$$

here expressed in radians ($180^\circ = \pi$), and as

$$a = q/(1 - e) \quad \text{and} \quad c_3 = (E - \sin E)/E^3 \quad ,$$

we have first

$$(1 - e) \cdot E + e \cdot c_3(E) \cdot E^3 = \sqrt{GM_{\odot} \left(\frac{1 - e}{q} \right)^3} \cdot (t - t_0) \quad ,$$

and, introducing a new variable

$$U = \sqrt{\frac{3e \cdot c_3(E)}{1 - e}} \cdot E$$

we obtain the modified form

$$U + \frac{1}{3}U^3 = \sqrt{6ec_3(E)} \cdot \sqrt{\frac{GM_\odot}{2q^3}} \cdot (t - t_0) \quad .$$

This equation has the same general form as Barker's equation and can be solved for U , the right side being known. With $c_1(E) = \sin(E)/E$ and $c_2(E) = (1 - \cos(E))/E^2$ we further obtain

$$\begin{aligned} x &= r \cos \nu = \frac{q}{1-e} (\cos E - e) &= q \cdot \left(1 - \left[\frac{2c_2}{6c_3}\right] U^2\right) \\ y &= r \sin \nu = \frac{q}{1-e} \sqrt{1-e^2} \sin E &= 2q \cdot \sqrt{\frac{1+e}{2e}} \cdot \left[\frac{1}{6c_3}\right] \cdot c_1 \cdot U \quad (4.16) \\ r & &= q \cdot \left(1 + \left[\frac{2c_2}{6c_3}\right] U^2\right) \quad . \end{aligned}$$

Later the equations for the velocity of a comet will be required, but their derivation is involved and will, therefore, be omitted here:

$$\begin{aligned} \dot{x} &= -\sqrt{\frac{GM_\odot}{q(1+e)}} \left(\frac{y}{r}\right) \\ \dot{y} &= +\sqrt{\frac{GM_\odot}{q(1+e)}} \left(\frac{x}{r} + e\right) \quad . \end{aligned} \quad (4.17)$$

In addition, the relationship between U and the true anomaly ν is

$$\tan\left(\frac{\nu}{2}\right) = \left[\sqrt{\frac{1+e}{3ec_3}} \cdot \frac{c_2}{c_1}\right] \cdot U \quad ,$$

which is mentioned here for the sake of completeness.

All the terms that are enclosed in square brackets in the preceding equations have the property of tending to a value of 1 for $E \rightarrow 0$ and $e \rightarrow 1$. In this limiting case, the equations reduce to the equations for a parabolic orbit, which have already been discussed.

These rearranged forms do not, of course, alter the fact that Kepler's equation must still be solved iteratively. What is important, however, is that the iteration no longer presents mathematical difficulties. We begin with the assumption that $E \approx 0$ ($c_3(E) \approx 1/6$) and determine the appropriate value for U by solving Kepler's/Barker's equation

$$U = B - 1/B$$

where

$$B = \sqrt[3]{A + \sqrt{A^2 + 1}} \quad \text{and} \quad A = \frac{3}{2} \sqrt{6ec_3(E)} \sqrt{\frac{GM_\odot}{2q^3}} \cdot (t - t_0) \quad .$$

From this we obtain improved values

$$\hat{E} = U \cdot \sqrt{\frac{1-e}{3e \cdot c_3(E)}} \quad \text{and} \quad c_3(\hat{E}) = \frac{\hat{E} - \sin \hat{E}}{\hat{E}^2},$$

with which we again obtain a more accurate value $\hat{u}$. These steps are repeated until U no longer varies beyond the desired degree of accuracy. Nevertheless it is important that the functions $c_1(E)$, $c_2(E)$ and $c_3(E)$ are correctly evaluated for small values of E . By virtue of the fact that individual terms cancel out, it is not possible to do this by directly evaluating the trigonometrical functions. Instead, we use the Taylor expansions of c_1 , c_2 and c_3 , which have the following form:

$$\begin{aligned} c_1(E) &= \frac{\sin E}{E} = 1 - \frac{E^2}{3!} + \frac{E^4}{5!} - \dots = \sum_{n=1}^{\infty} \alpha_n \\ c_2(E) &= \frac{1 - \cos E}{E^2} = \frac{1}{2!} - \frac{E^2}{4!} + \frac{E^4}{6!} - \dots = \sum_{n=1}^{\infty} \beta_n \\ c_3(E) &= \frac{E - \sin E}{E^3} = \frac{1}{3!} - \frac{E^2}{5!} + \frac{E^4}{7!} - \dots = \sum_{n=1}^{\infty} \gamma_n \end{aligned} \quad (4.18)$$

All three series contain even powers of E exclusively, and therefore converge very rapidly. The summands can conveniently be obtained by the following recursive procedure:

$$\begin{aligned} \alpha_1 &= 1 \\ \beta_n &= \alpha_n \cdot \frac{1}{2n}, \quad \gamma_n = \beta_n \cdot \frac{1}{2n+1}, \quad \alpha_{n+1} = -E^2 \cdot \gamma_n \quad (n = 1, \dots) \end{aligned}$$

This process is carried out in the STUMPPFF procedure until a summand becomes smaller than the required accuracy EPS.

```
(*-----*)
(* STUMPPFF: calculation of Stumpff's functions C1 = sin(E)/E ,      *)
(* C2 = (1-cos(E))/(E**2) and C3 = (E-sin(E))/(E**3)                *)
(* for argument E2=E**2                                              *)
(* (E: eccentric anomaly in radians)                                  *)
(*-----*)
PROCEDURE STUMPPFF(E2:REAL;VAR C1,C2,C3:REAL);
CONST EPS=1E-12;
VAR N,ADD: REAL;
BEGIN
  C1:=0.0; C2:=0.0; C3:=0.0; ADD:=1.0; N:=1.0;
  REPEAT
    C1:=C1+ADD; ADD:=ADD/(2.0*N);
    C2:=C2+ADD; ADD:=ADD/(2.0*N+1.0);
    C3:=C3+ADD; ADD:=-E2*ADD; N:=N+1.0;
  UNTIL ABS(ADD)<EPS;
END;
(*-----*)
```


The remaining equations for parabolic and near-parabolic cometary orbits are coded in routine PARAB, which is called in the same way as HYPERB.

```
(*-----*)
(* PARAB: calculation of position and velocity for *)
(* parabolic and near-parabolic orbits according to Stumpff *)
(* *)
(* T0 time of perihelion passage X,Y position *)
(* T time VI,VY velocity *)
(* Q perihelion distance *)
(* ECC eccentricity *)
(* (T0,T in julian centuries since J2000) *)
(*-----*)
PROCEDURE PARAB(T0,T,Q,ECC:REAL;VAR X,Y,VI,VY:REAL);
  CONST EPS = 1E-9;
  KGAUSS = 0.01720209895;
  MAXIT = 15;
  VAR E2,E20,FAC,C1,C2,C3,K,TAU,A,U,U2: REAL;
  R: REAL;
  I: INTEGER;
  BEGIN
    E2:=0.0; FAC:=0.5*ECC; I:=0;
    K := KGAUSS / SQRT(Q*(1.0+ECC));
    TAU := KGAUSS * 36525.0*(T-T0);
    REPEAT
      I:=I+1;
      E20:=E2;
      A:=1.5*SQRT(FAC/(Q*Q*Q))*TAU; A:=CUBR(SQRT(A*A+1.0)+A);
      U:=A-1.0/A; U2:=U*U; E2:=U2*(1.0-ECC)/FAC;
      STUMPPFF(E2,C1,C2,C3); FAC:=3.0*ECC*C3;
    UNTIL (ABS(E2-E20)<EPS)OR(I>MAXIT);
    IF (I=MAXIT) THEN WRITELN(' convergence problems in PARAB');
    R :=Q*(1.0+U2*C2+ECC/FAC);
    X :=Q*(1.0-U2*C2/FAC); VY:= K*(X/R+ECC);
    Y :=Q*SQRT((1.0+ECC)/FAC)*U*C1; VI:=-K*Y/R;
  END;
(*-----*)
```

4.5 Gaussian Vectors

The various procedures described so far enable us to calculate the position of a comet in the plane of its orbit. In order to be able to determine the ecliptic coordinates of the comet, we must first establish the orientation of the orbit relative to the ecliptic. For this we use three additional orbital elements (Fig. 4.6):

- i The *inclination* gives the angle of intersection between the orbital plane and the ecliptic. An inclination of more than 90° means that the comet is retrograde, its direction of revolution around the Sun being opposite to that of the planets.

Ω The *longitude of the ascending node* indicates the angle between the vernal equinox and the point on the orbit at which the comet crosses the ecliptic from south to north.

ω The *argument of perihelion* is the angle between the direction of the ascending node and the direction of the closest point of the orbit to the Sun.

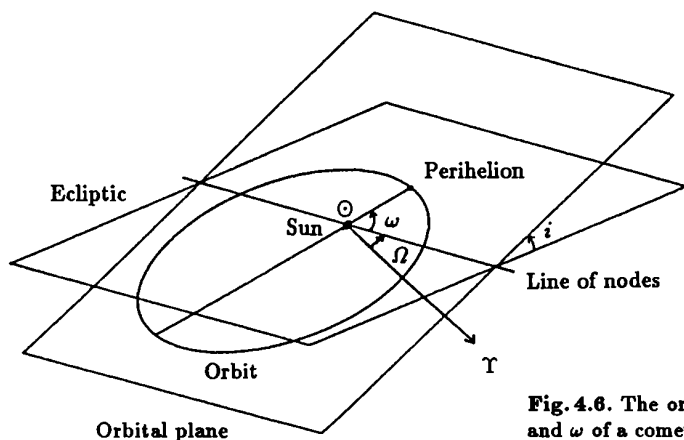


Fig. 4.6. The orbital elements i , Ω and ω of a comet

Frequently, the last element ω , is replaced by the *longitude of perihelion* ($\varpi = \Omega + \omega$), defined as being the sum of the longitude of the ascending node and the argument of perihelion.

Because the position of the vernal equinox and the ecliptic are slowly altering as a result of precession, the equinox is given as well as the orbital elements. 'Equinox 1950', for example, means that the elements are referred to the 1950 vernal equinox and ecliptic.

In order to determine the transformation equations, we employ a coordinate system (x'' , y'' , z''), where the x'' -axis coincides with the ascending node, and the x'' - y'' plane with the orbital plane (Fig. 4.7). A point on the orbit with true anomaly ν and distance r from the Sun has the following coordinates in this system:

$$\begin{aligned} x'' &= r \cos u \\ y'' &= r \sin u \\ z'' &= 0 \end{aligned} ,$$

where the *argument of latitude* u is the sum of the argument of perihelion and the true anomaly:

$$u = \omega + \nu \quad . \quad (4.19)$$

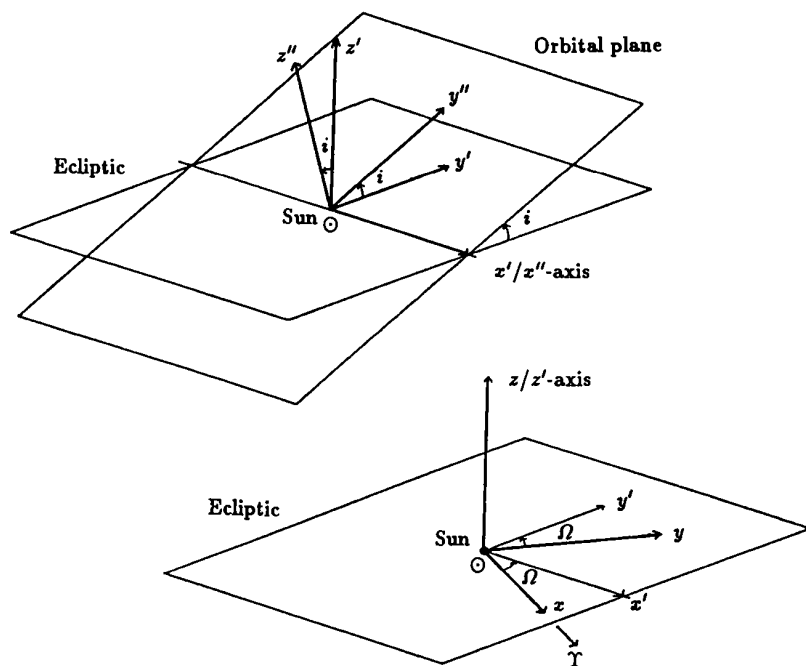


Fig. 4.7. Conversion to ecliptic coordinates

We now consider the (x', y', z') system, where the x' -axis also indicates the ascending node, and where the x' - y' plane coincides with the ecliptic. Both coordinate systems are simply inclined to one another at the orbital inclination i . Therefore:

$$\begin{aligned} x' &= x'' &= r \cos u \\ y' &= y'' \cos i - z'' \sin i &= r \sin u \cos i \\ z' &= y'' \sin i + z'' \cos i &= r \sin u \sin i \end{aligned}$$

The coordinates thus obtained refer to the ecliptic, but are reckoned from the ascending node. These must therefore be altered in the next step into yet another coordinate system, where the (x, y, z) axes are the conventional system, referred to the vernal equinox and the ecliptic:

$$\begin{aligned} x &= x' \cos \Omega - y' \sin \Omega \\ y &= x' \sin \Omega + y' \cos \Omega \\ z &= z' \end{aligned}$$

We therefore obtain the combined equations

$$\begin{aligned} x &= r \cdot (\cos u \cos \Omega - \sin u \cos i \sin \Omega) \\ y &= r \cdot (\cos u \sin \Omega + \sin u \cos i \cos \Omega) \\ z &= r \cdot (\sin u \sin i) \end{aligned} \quad (4.20)$$

Using the addition theorems, we may replace

$$\begin{aligned}\cos u & \text{ by } \cos \omega \cos \nu - \sin \omega \sin \nu \quad \text{and} \\ \sin u & \text{ by } \sin \omega \cos \nu + \cos \omega \sin \nu \quad ,\end{aligned}$$

to obtain a formulation first suggested by Gauss that is particularly suitable for practical calculations:

$$\begin{aligned}x &= r \cos \nu \cdot P_x + r \sin \nu \cdot Q_x \\ y &= r \cos \nu \cdot P_y + r \sin \nu \cdot Q_y \\ z &= r \cos \nu \cdot P_z + r \sin \nu \cdot Q_z\end{aligned}\tag{4.21}$$

where

$$\begin{aligned}P_x &= +\cos \omega \cos \Omega - \sin \omega \cos i \sin \Omega \\ P_y &= +\cos \omega \sin \Omega + \sin \omega \cos i \cos \Omega \\ P_z &= +\sin \omega \sin i\end{aligned}\tag{4.22}$$

and

$$\begin{aligned}Q_x &= -\sin \omega \cos \Omega - \cos \omega \cos i \sin \Omega \\ Q_y &= -\sin \omega \sin \Omega + \cos \omega \cos i \cos \Omega \\ Q_z &= +\cos \omega \sin i \quad .\end{aligned}\tag{4.23}$$

The variables (P_x, P_y, P_z) and (Q_x, Q_y, Q_z) may be taken as the components of two vectors $\mathbf{P}$ and $\mathbf{Q}$, known as the Gaussian vectors, which both lie in the plane of the orbit, are at right-angles to one another and have unit length. $\mathbf{P}$ points towards perihelion. $\mathbf{P}$ and $\mathbf{Q}$ thus form the x - and y -axes of the coordinate system in which we expressed the orbital motion of the comet, as covered in the preceding sections. Using $(\tilde{x}, \tilde{y}) = (r \cos \nu, r \sin \nu)$ and $(\dot{\tilde{x}}, \dot{\tilde{y}})$ to distinguish between the position and velocity expressed in these coordinates and those in ecliptic coordinates ($\mathbf{r} = (x, y, z)$ and $\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})$), we have

$$\begin{aligned}\mathbf{r} &= \tilde{x}\mathbf{P} + \tilde{y}\mathbf{Q} \\ \mathbf{v} &= \dot{\tilde{x}}\mathbf{P} + \dot{\tilde{y}}\mathbf{Q} \quad .\end{aligned}\tag{4.24}$$

The use of these relationships is particularly recommended when we want to calculate a whole series of heliocentric cometary positions at a time. The Gaussian vectors then only have to be determined once at the beginning, as they only depend on the orbital elements and not on time.

```
(*-----*)
(* GAUSVEC: calculation of the Gaussian vectors (P,Q,R) from      *)
(*      ecliptic orbital elements:                                *)
(*      LAN = longitude of the ascending node                      *)
(*      INC = inclination                                           *)
(*      AOP = argument of perihelion                               *)
(*-----*)
```

```

PROCEDURE GAUSVEC(LAN,INC,AOP:REAL;VAR PQR:REAL33);
  VAR C1,S1,C2,S2,C3,S3: REAL;
  BEGIN
    C1:=CS(AOP); C2:=CS(INC); C3:=CS(LAN);
    S1:=SN(AOP); S2:=SN(INC); S3:=SN(LAN);
    PQR[1,1]:=+C1*C3-S1*C2*S3; PQR[1,2]:=-S1*C3-C1*C2*S3; PQR[1,3]:=+S2*S3;
    PQR[2,1]:=+C1*S3+S1*C2*C3; PQR[2,2]:=-S1*S3+C1*C2*C3; PQR[2,3]:=-S2*C3;
    PQR[3,1]:=+S1*S2; PQR[3,2]:=+C1*S2; PQR[3,3]:=+C2;
  END;
(*-----*)
(* ORBECL: transformation of coordinates XX,YY referred to the *)
(* orbital plane into ecliptic coordinates X,Y,Z using *)
(* Gaussian vectors PQR *)
(*-----*)
PROCEDURE ORBECL(XX,YY:REAL;PQR:REAL33;VAR X,Y,Z:REAL);
  BEGIN
    X:=PQR[1,1]*XX+PQR[1,2]*YY;
    Y:=PQR[2,1]*XX+PQR[2,2]*YY;
    Z:=PQR[3,1]*XX+PQR[3,2]*YY;
  END;
(*-----*)

```

The GAUSVEC procedure uses the elements of the orbit to calculate the vectors P and Q , as well as the vector R , which is perpendicular to P and Q , and like these has unit length. Although R is not absolutely essential, it is simple to calculate, and is useful for other purposes. PQR is a 3×3 -matrix, which is declared as data type REAL33. A corresponding definition

```
TYPE REAL33: ARRAY[1..3,1..3];
```

has to be included in the relevant main program. We are now able to use the routine ORBECL to determine the three-dimensional coordinates of a comet relative to the ecliptic from the corresponding Cartesian coordinates in the orbital plane.

The various routines that have been developed may be combined into an overall procedure that is suitable for all types of orbit, and which yields heliocentric ecliptic coordinates directly.

```

(*-----*)
(* KEPLER: calculation of position and velocity for unperturbed *)
(* elliptic, parabolic and hyperbolic orbits *)
(* *)
(* T0 time of perihelion passage X,Y,Z position *)
(* T time VX,VY,VZ velocity *)
(* Q perihelion distance *)
(* ECC eccentricity *)
(* PQR matrix of Gaussian vectors *)
(* (T0,T in Julian centuries since J2000) *)
(*-----*)
PROCEDURE KEPLER(T0,T,Q,ECC:REAL;PQR:REAL33;VAR X,Y,Z,VX,VY,VZ:REAL);
  CONST MO=5.0; EPS=0.1;
  KGAUSS = 0.01720209895; DEG = 57.29577951;

```

```

VAR M,DELTA,TAU,INVAX,XX,YY,VVX,VVY: REAL;
BEGIN
  DELTA := ABS(1.0-ECC);
  INVAX := DELTA / Q;
  TAU := KGAUSS*36525.0*(T-T0);
  M := DEG*TAU*SQRT(INVAX*INVAX*INVAX);
  IF ( (M<M0) AND (DELTA<EPS) )
    THEN PARAB(T0,T,Q,ECC,XX,YY,VVX,VVY)
    ELSE IF (ECC<1.0)
      THEN ELLIP (M,1.0/INVAX,ECC,XX,YY,VVX,VVY)
      ELSE HYPERB(T0,T,1.0/INVAX,ECC,XX,YY,VVX,VVY);
  ORBECL(XX,YY,PQR,X,Y,Z); ORBECL(VVX,VVY,PQR,VX,VY,VZ);
END;
(*-----*)

```

4.6 Light-Time

The light observed from a comet requires a certain amount of time to reach us. Depending on the comet's distance, this time may range from a few minutes to a few hours. As the comet continues to move during this light-time, the observed position of the comet is slightly behind the actual geometrical position. The velocity of light is significantly greater than the typical velocity of a comet, which means that taking light-time into account generally involves a small correction of the order of only a few arc-seconds to about one arc-minute. We can therefore use a few approximations, which allow us to deal with light-time relatively simply. In order to calculate the position $\mathbf{r}(t)$ at which the comet will be seen at time t , we first calculate the comet's heliocentric coordinates $\mathbf{r}_k(t)$ and the geocentric coordinates of the Sun $\mathbf{r}_\odot(t)$. The geometrical distance of the comet from the Earth at time t is then given by

$$\Delta_0 = |\mathbf{r}_0| = |\mathbf{r}_\odot(t) + \mathbf{r}_k(t)| = \sqrt{(x_\odot + x_k)^2 + (y_\odot + y_k)^2 + (z_\odot + z_k)^2}.$$

It differs only slightly from the distance actually traversed by a ray of light, which is

$$\Delta = |\mathbf{r}_\odot(t) + \mathbf{r}_k(t')| \text{ where } t' = t - \frac{\Delta}{c},$$

but which can only be calculated iteratively, because the point in time t' at which the light is emitted by the comet cannot be known in advance. Light-time is therefore $\tau \approx \Delta_0/c$, and the heliocentric position of the comet at time t' can be obtained to a good approximation from

$$\mathbf{r}_k(t') \approx \mathbf{r}_k(t) - \mathbf{v}_k(t) \cdot \frac{\Delta_0}{c}.$$

Using these *retarded* heliocentric coordinates of the comet, the observed, geocentric position vector is

$$\mathbf{r} \approx \mathbf{r}_\odot(t) + \left\{ \mathbf{r}_k(t) - \mathbf{v}_k(t) \cdot \frac{\Delta_0}{c} \right\} . \quad (4.25)$$

With the units employed here (AU and d), the factor $1/c$ has a value of

$$1/c = 0^d.00578/\text{AU} .$$

Cometary coordinates corrected in the manner described are known as *astrometric* coordinates. They are suitable for directly comparing the position of a comet with stellar coordinates taken from a star atlas. It should be noted that it is customary to give the geocentric distance as the value of the geometrical distance Δ_0 (and not of the light-path Δ).

4.7 The COMET Program

The COMET program calculates ephemerides of comets and minor planets on the basis of the two-body problem. Perturbations of the body under consideration by other planets are not taken into account. This simplification generally suffices for most practical applications. In comparing the results given by this program and those from other sources, however, we must remember that the data given in the latter may not have been calculated with the same assumptions. The user does not have to bother with the type of orbit (ellipse, parabola, or hyperbola). COMET recognizes this from the given eccentricity and chooses the appropriate method of calculation. Elongated, high eccentricity ellipses are handled by the mathematically sound method devised by Stumpff.

COMET calculates heliocentric ecliptic and geocentric equatorial coordinates. In both cases, precession to the required equinox is taken into account. In addition, the geocentric coordinates are corrected for light-time effects. Geocentric coordinates in this form (including precession and light-time effects) are designated astrometric coordinates. They are directly comparable with positions in a star catalogue, or may be used to plot the position of the object on a star chart with the same equinox!

In comparing the results with other sources, it is important to check whether the latter do really give astrometric coordinates with the same equinox, or even if perhaps they are apparent coordinates (in which nutation and aberration are also taken into account).

```
(*-----*)
(*                      COMET                      *)
(* calculation of unperturbed ephemeris with arbitrary eccentricity *)
(* for comets and minor planets                      *)
(*                      version 1993/07/01          *)
(*-----*)
```

```
PROGRAM COMET(INPUT,OUTPUT,COMINP);
```

```
TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
```

```
VAR DAY,MONTH,YEAR,NLINE           : INTEGER;
    D,HOUR,TO,Q,ECC,TEQX0,FAC      : REAL;
    MODJD,T,T1,DT,T2,TEQX         : REAL;
    X,Y,Z,VX,VY,VZ,XS,YS,ZS       : REAL;
    L,B,R,LS,BS,RS,RA,DEC,DELTA,DELTA0 : REAL;
    PQR,A,AS                       : REAL33;
    COMINP                         : TEXT;
```

```
(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS; ASN, ATN, ATN2, CART, POLAR, DMS, CUBR                      *)
(* MJD, CALDAT                                                            *)
(* ECLEQU, GAUSVEC, ORBECL                                              *)
(* PMATECL, PRECART                                                    *)
(* ECCANOM, ELLIP, HYBANOM, HYPERB, STUMPPFF, PARAB, KEPLER           *)
(* SUN200                                                                *)
(*-----*)
```

```
(*-----*)
(* GETELM: reads orbital elements from file COMINP                      *)
(*-----*)
```

```
PROCEDURE GETELM (VAR TO,Q,ECC:REAL;VAR PQR:REAL33;VAR TEQX0:REAL);
```

```
VAR INC,LAN,AOP: REAL;
```

```
BEGIN
```

```
WRITELN;
WRITELN (' COMET: ephemeris calculation for comets and minor planets');
WRITELN ('                      version 93/07/01                      ');
WRITELN ('                      (C) 1993 Thomas Pfleger, Oliver Montenbruck          ');
WRITELN;
WRITELN (' Orbital elements from file COMINP: ');
WRITELN;
```

```
(* open file for reading *)
(* RESET(COMINP); *) (* Standard Pascal *)
ASSIGN(COMINP,'COMINP.DAT'); RESET(COMINP); (* TURBO Pascal *)
(* RESET(COMINP,'COMINP.DAT'); *) (* ST Pascal plus *)
```



```
(* display orbital elements *)
READLN (COMINP, YEAR, MONTH, D);
WRITELN(' perihelion time (y m d) ', YEAR:7, MONTH:3, D:6:2);
READLN (COMINP, Q);
WRITELN(' perihelion distance (q) ', Q:14:7, ' AU ');
READLN (COMINP, ECC);
WRITELN(' eccentricity (e) ', ECC:14:7);
READLN (COMINP, INC);
WRITELN(' inclination (i) ', INC:12:5, ' deg');
READLN (COMINP, LAN);
WRITELN(' long. of ascending node ', LAN:12:5, ' deg');
READLN (COMINP, AOP);
WRITELN(' argument of perihelion ', AOP:12:5, ' deg');
READLN (COMINP, TEQX0);
WRITELN(' equinox ', TEQX0:9:2);

DAY:=TRUNC(D); HOUR:=24.0*(D-DAY);
TO := ( MJD(DAY, MONTH, YEAR, HOUR) - 51544.5) / 36525.0;
TEQX0 := (TEQX0-2000.0)/100.0;
GAUSVEC(LAN, INC, AOP, PQR);
```

END;

```
(*-----*)
(* GETEPH: reads desired period of time and equinox of the ephemeris *)
(*-----*)
PROCEDURE GETEPH(VAR T1, DT, T2, TEQX:REAL);
```

```
VAR YEAR, MONTH, DAY : INTEGER;
    EQX, HOUR, JD : REAL;
```

BEGIN

```
WRITELN;
WRITELN(' Begin and end of the ephemeris: ');
WRITELN;
WRITE (' first date (yyyy mm dd hh.hhh) ... ');
READLN (YEAR, MONTH, DAY, HOUR);
T1 := ( MJD(DAY, MONTH, YEAR, HOUR) - 51544.5) / 36525.0;
WRITE (' final date (yyyy mm dd hh.hhh) ... ');
READLN (YEAR, MONTH, DAY, HOUR);
T2 := ( MJD(DAY, MONTH, YEAR, HOUR) - 51544.5) / 36525.0;
WRITE (' step size (dd hh.hh) ... ');
READLN (DAY, HOUR);
DT := ( DAY + HOUR/24.0 ) / 36525.0;
WRITELN;
WRITE (' Desired equinox of the ephemeris (yyyy.y) ... ');
READLN (EQX);
TEQX := (EQX-2000.0)/100.0;
WRITELN; WRITELN;
WRITELN (' Date ET Sun l b r',
        ' Ra Dec Distance ');
WRITELN (' :45, ' h m s o ' ' (AU) ');
```

END;

```

(*-----*)
(* WRTLBR: write L and B in deg,min,sec and R *)
(*-----*)
PROCEDURE WRTLBR(L,B,R:REAL);
  VAR H,M: INTEGER;
      S : REAL;
  BEGIN
    DMS(L,H,M,S); WRITE (H:5,M:3,S:5:1);
    DMS(B,H,M,S); WRITELN(H:5,M:3,TRUNC(S+0.5):3,R:11:6);
  END;

(*-----*)

BEGIN (* COMET *)

  GETELM (T0,Q,ECC,PQR,TEQX); (* read orbital elements *)
  GETEPH (T1,DT,T2,TEQX); (* read period of time and equinox *)

  NLINE := 0;

  PMATECL (TEQX,TEQX,A); (* calculate precession matrix *)

  T := T1;

  REPEAT

    (* date *)

    MODJD := T*36525.0+51544.5; CALDAT (MODJD,DAY,MONTH,YEAR,HOURL);

    (* ecliptic coordinates of the sun, equinox TEQX *)

    SUN200 (T,LS,BS,RS); CART (RS,BS,LS,XS,YS,ZS);
    PMATECL (T,TEQX,AS); PRECART (AS,XS,YS,ZS);

    (* heliocentric ecliptic coordinates of the comet *)

    KEPLER (T0,T,Q,ECC,PQR,X,Y,Z,VX,VY,VZ);
    PRECART (A,X,Y,Z); PRECART (A,VX,VY,VZ); POLAR (X,Y,Z,R,B,L);

    (* geometric geocentric coordinates of the comet *)

    X:=X+XS; Y:=Y+YS; Z:=Z+ZS; DELTA0 := SQRT ( X*X + Y*Y + Z*Z );

    (* first order correction for light travel time *)

    FAC:=0.00578*DELTA0; X:=X-FAC*VX; Y:=Y-FAC*VY; Z:=Z-FAC*VZ;
    ECLEQU (TEQX,X,Y,Z); POLAR (X,Y,Z,DELTA,DEC,RA); RA:=RA/15.0;

```

```
(* output *)
```

```
WRITE(YEAR:4,'/',MONTH:2,'/',DAY:2,HOOR:6:1);
WRITE(LS:7:1,L:7:1,B:6:1,R:7:3); WRTLBR(RA,DEC,DELTA0);
NLINE := NLINE+1; IF (NLINE MOD 5) = 0 THEN WRITELN;
```

```
(* next time step
```

```
*)
```

```
T := T + DT;
```

```
UNTIL (T2<T);
```

```
END.
```

```
(*-----*)
```

As an example for the use of COMET we will calculate an ephemeris for Comet Halley for its period of visibility during its last return. First, the orbital elements should be entered into a data file with the name COMINP.DAT. The comments (the exclamation marks and succeeding words on each line) are only there for information, and may be omitted if not required. For Comet Halley at its 1985/1986 return the details are:

```
1986 2 9.43867 ! Time of perihelion (year month day.fraction)
0.5870992 ! Perihelion distance q in AU
0.9672725 ! Eccentricity e
162.23932 ! Inclination i
58.14397 ! Longitude of the ascending node
111.84658 ! Argument of perihelion
1950.0 ! Equinox for the orbital elements
```

When run, COMET displays the data read from the data file COMINP.DAT as a check, before asking the user to enter the time interval, the step width and the required equinox:

```
COMET: ephemeris calculation for comets and minor planets
      version 93/07/01
      (C) 1993 Thomas Pfleger, Oliver Montenbruck
```

```
Orbital elements from file COMINP:
```

```
perihelion time (y m d) 1986 2 9.44
perihelion distance (q) 0.5870992 AU
eccentricity (e) 0.9672725
inclination (i) 162.23932 deg
long. of ascending node 58.14397 deg
argument of perihelion 111.84658 deg
equinox 1950.00
```

We want an ephemeris for the period between 1985 November 15 and 1986 April 1, in ten-day steps. We choose 2000 for the equinox. The extreme dates

for the ephemeris are to be given in the form: Year, Month, Day and Hour with decimal fraction. User input is shown in *italic*.

Begin and end of the ephemeris:

```
first date (yyyy mm dd hh.hhh)      ... 1985 11 15 0.0
final date (yyyy mm dd hh.hhh)      ... 1986 04 01 0.0
step size (dd hh.hh)                 ... 10 00.0
```

Desired equinox of the ephemeris (yyyy.y) ... 2000.0

COMET now calculates the ephemeris for the data entered:

Date	ET	Sun	l	b	r	RA	Dec	Distance
						h m s	o ' "	(AU)
1985/11/15	0.0	232.6	56.9	0.6	1.720	4 0 40.3	22 4 28	0.736822
1985/11/25	0.0	242.7	53.2	1.8	1.572	2 15 19.1	18 24 12	0.623053
1985/12/ 5	0.0	252.8	48.7	3.2	1.422	0 28 47.5	10 39 0	0.665665
1985/12/15	0.0	263.0	43.1	5.0	1.269	23 18 18.6	3 53 22	0.821652
1985/12/25	0.0	273.1	35.9	7.1	1.114	22 36 56.1	0-20 5	1.019722
1986/ 1/ 4	0.0	283.3	26.2	9.8	0.961	22 10 36.0	-3 0 26	1.216950
1986/ 1/14	0.0	293.5	12.8	13.0	0.814	21 50 58.9	-4 58 44	1.388260
1986/ 1/24	0.0	303.7	353.4	16.2	0.687	21 33 26.2	-6 47 36	1.511970
1986/ 2/ 3	0.0	313.9	326.3	17.7	0.604	21 15 33.4	-8 49 26	1.563479
1986/ 2/13	0.0	324.0	294.8	14.9	0.592	20 57 11.7	-11 15 58	1.520206
1986/ 2/23	0.0	334.1	267.2	8.7	0.658	20 39 33.7	-14 9 18	1.383203
1986/ 3/ 5	0.0	344.1	247.1	2.6	0.775	20 22 12.2	-17 39 45	1.179038
1986/ 3/15	0.0	354.1	232.8	-1.9	0.918	20 0 46.1	-22 28 16	0.936978
1986/ 3/25	0.0	4.1	222.5	-5.2	1.070	19 22 34.1	-30 13 41	0.685468

The first two columns contain date and time. The column headed Sun then gives the geocentric ecliptic longitude of the Sun. One can use this information to estimate the elongation of the comet (or minor planet) from the Sun. This gives a clue as to whether the object is accessible for observation. The three following columns (headed l, b, and r) give the heliocentric ecliptic coordinates of the body. The last three columns give the geocentric coordinates right ascension and declination, and the geocentric, geometric distance of the comet in astronomical units (not corrected for light-time.)

5. Special Perturbations

The calculation of cometary and planetary orbits, based on a solution of the unperturbed two-body problem, offers a simple and easily understood method of predicting the position of such a body at any desired time. In addition, the analytical description of the motion in terms of conic sections also provides a particularly clear way of describing the relationship between the various orbital elements and the position in the orbit.

There is, however, the disadvantage that the perturbations caused by the major planets cannot be incorporated. Although the gravitational attraction exerted by the planets is generally an order of magnitude smaller than that of the Sun, over a period of many years it can lead to considerable deviations from an unperturbed orbit. The perturbations are particularly significant when a comet or a minor planet repeatedly passes close to one of the major planets. In this case, substantial changes are possible – as may be observed with various periodic comets. Examples are Comet P/Pons-Winnecke (1819 III), whose orbital inclination increased from 10° to 22° between 1819 and 1976 because its aphelion lay close to Jupiter, and Comet P/Wolf (1884 III), whose orbital period increased from 6.8 years to 8.3 years, following an encounter with Jupiter in 1922.

Although the perturbations caused by the major planets to most minor-planet orbits are far smaller than those mentioned in these examples, they still generally prevent satisfactory predictions of the orbits over a period of many years on the basis of a simple Keplerian orbit. This means that to find and identify faint minor planets it is essential to take these perturbations into account in calculating the ephemerides. Basically, there are two different procedures that may be used to do this. In the *special perturbations* method, we begin with the position and velocity of the minor planet at an initial epoch, and calculate the orbit step by step, using numerical integration, taking the accelerations caused by the Sun and all the planets into account, until we reach the desired end-point. The second possibility, the *general perturbations* method, consists of expressing the deviations from an unperturbed orbit as a periodic series, which then enables us to calculate the perturbed position at any given time. The advantage of an analytical method is, however, offset by having to devote considerable effort to developing the series expansions, which prevents the method from being applied to the several thousand minor planets. Analytical perturbation theories are therefore generally available for just the major planets (see Chap. 6).

5.1 Equation of Motion

The basis for describing a minor planet's orbit using numerical integration is what is known as the equation of motion, which expresses the acceleration as a function of position and time. According to Newton's law of gravity, a body of mass m in the gravitational field of just the Sun is subject to a force

$$F = ma = GM_{\odot}m \frac{1}{r^2} \quad , \quad (5.1)$$

which decreases as the square of the distance r from the Sun. Here, once again (see Chap. 4),

$$GM_{\odot} = k^2 \text{AU}^3 \text{d}^{-2} \quad \text{where} \quad k = 0.01720209895$$

is the product of the gravitational constant and the mass of the Sun. Taking into account the fact that the force is always directed towards the Sun, we find that the resultant acceleration, expressed in terms of the corresponding unit vector $-\mathbf{r}/r$ is

$$\mathbf{a} = -GM_{\odot} \frac{\mathbf{r}}{r^3} \quad . \quad (5.2)$$

As we can see, the acceleration is not dependent on the mass m of the body, because the gravitational force itself increases as m .

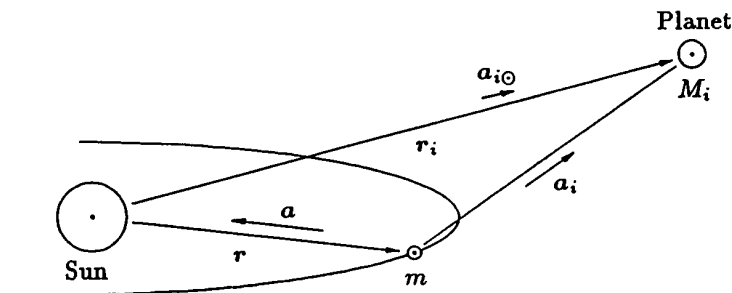


Fig. 5.1. Acceleration of a body m by the Sun and planets

Similarly, the gravitational force of a planet of mass M_i at point $\mathbf{r}_i$ gives rise to an additional acceleration

$$\mathbf{a}_i = -GM_i \frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^3} \quad (5.3)$$

of the body under consideration. However, because the planet also simultaneously imposes an acceleration of magnitude

$$\mathbf{a}_{i\odot} = -GM_i \frac{-\mathbf{r}_i}{|\mathbf{r}_i|^3} \quad (5.4)$$

on the Sun, the overall acceleration acting on the mass m is

$$\Delta \mathbf{a}_i = \mathbf{a}_i - \mathbf{a}_{i\odot} = -GM_i \left(\frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^3} + \frac{\mathbf{r}_i}{|\mathbf{r}_i|^3} \right) \quad (5.5)$$

relative to the Sun. Here, the first term is also known as the direct perturbation acceleration, and the second as the indirect acceleration.

If the contributions of all nine planets are taken together, then the following equation gives the overall acceleration of a comet or minor planet:

$$\mathbf{a} = -GM_\odot \frac{\mathbf{r}}{r^3} - \sum_{i=1}^9 GM_i \left(\frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^3} + \frac{\mathbf{r}_i}{|\mathbf{r}_i|^3} \right) \quad (5.6)$$

Table 5.1 gives the ratios of the masses of the Sun and the planets and from this we can see that, in general, even the largest planets, Jupiter and Saturn, will produce only minor perturbations. If, however, the distance of these planets is less than 1/10 AU, and certainly if less than 1/100 AU, then the corresponding acceleration may be of the same order of magnitude as that caused by the Sun's own gravitational attraction.

Table 5.1. Ratio of planetary masses to that of the Sun ($M_\odot$) (IAU 1976)

Planet	mass ratio
Mercury	$M_\odot/M_1 = 6023600.0$
Venus	$M_\odot/M_2 = 408523.5$
Earth/Moon	$M_\odot/M_3 = 328900.5$
Mars	$M_\odot/M_4 = 3098710.0$
Jupiter	$M_\odot/M_5 = 1047.355$
Saturn	$M_\odot/M_6 = 3498.5$
Uranus	$M_\odot/M_7 = 22869.0$
Neptune	$M_\odot/M_8 = 19314.0$
Pluto	$M_\odot/M_9 = 3000000.0$

The acceleration $\mathbf{a} = (a_x, a_y, a_z)$ acting on a body at heliocentric coordinates $\mathbf{r} = (x, y, z)$ at a given time

$$T = \frac{\text{JD} - 2451545}{36525} = \frac{\text{MJD} - 51544.5}{36525}$$

may be calculated, using the ACCEL subroutine. This assumes that there is an appropriate procedure

```
PROCEDURE POSITION ( PLANET: PLANET_TYPE; T: REAL; VAR X,Y,Z: REAL );
```

where

```
TYPE PLANET_TYPE = ( MERCURY, VENUS, EARTH, MARS,
                     JUPITER, SATURN, URANUS, NEPTUNE, PLUTO );
```

available to provide the corresponding, heliocentric, planetary coordinates. In principle, there is freedom to choose any coordinate system, but in what follows, we assume that the coordinates are based on the J2000 ecliptic and vernal equinox.

```
(*-----*)
(* ACCEL: computes the acceleration *)
(* T Time in Julian centuries since J2000 *)
(* X,Y,Z Heliocentric ecliptic coordinates (in AU) *)
(* AX,AY,AZ Acceleration (in AU/d**2) *)
(*-----*)
PROCEDURE ACCEL ( T, X,Y,Z: REAL; VAR AX,AY,AZ: REAL );
CONST K_GAUSS = 0.01720209895; (* Gaussian gravitational constant *)
VAR PLANET : PLANET_TYPE;
    GM_SUN, R_SQR,R,MU_R3 : REAL;
    XP,YP,ZP, DX,DY,DZ : REAL;
    MU : ARRAY[PLANET_TYPE] OF REAL;
BEGIN
    (* Grav. constant * solar and planetary masses in AU**3/d**2 *)
    GM_SUN := K_GAUSS*K_GAUSS;
    MU[MERCURY] := GM_SUN / 6023600.0;
    MU[VENUS] := GM_SUN / 408523.5;
    MU[EARTH] := GM_SUN / 328900.5;
    MU[MARS] := GM_SUN / 3098710.0;
    MU[JUPITER] := GM_SUN / 1047.355;
    MU[SATURN] := GM_SUN / 3498.5;
    MU[URANUS] := GM_SUN / 22869.0;
    MU[NEPTUNE] := GM_SUN / 19314.0;
    MU[PLUTO] := GM_SUN / 3000000.0;
    (* Solar attraction *)
    R_SQR := ( X*X + Y*Y + Z*Z ); R := SQRT(R_SQR);
    MU_R3 := GM_SUN / (R_SQR*R);
    AX := -MU_R3*X; AY := -MU_R3*Y; AZ := -MU_R3*Z;
    (* Planetary perturbation *)
    FOR PLANET := MERCURY TO PLUTO DO
        BEGIN
            (* Planetary coordinates *)
            POSITION ( PLANET, T, XP,YP,ZP );
            DX:=X-XP; DY:=Y-YP; DZ:=Z-ZP;
            (* Direct acceleration *)
            R_SQR := ( DX*DX + DY*DY + DZ*DZ ); R := SQRT(R_SQR);
            MU_R3 := MU[PLANET] / (R_SQR*R);
            AX := AX-MU_R3*DX; AY := AY-MU_R3*DY; AZ := AZ-MU_R3*DZ;
            (* Indirect acceleration *)
            R_SQR := ( XP*XP + YP*YP + ZP*ZP ); R := SQRT(R_SQR);
            MU_R3 := MU[PLANET] / (R_SQR*R);
```



```

AX := AX-MU_R3*XP;  AY := AY-MU_R3*YP;  AZ := AZ-MU_R3*ZP;
END;
END;
(*-----*)

```

Table 5.2. Planetary orbital elements for the period 1990-2000 relative to the J2000 ecliptic and vernal equinox. The Epoch t_0 is 2000 January 1.5 (JD 2451545.0) for the inner planets Mercury to Mars, and 1995 October 10.0 (JD 2450000.5) for the planets Jupiter to Pluto.

Planet	a (AE)	e	M	n	Ω	i	ω
Mercury	0.387099	0.205634	174°794	149472°6738/cy	48°331	7°004	77°455
Venus	0.723332	0.006773	50°407	58517°8149/cy	76°680	3°394	131°571
Earth	1.000000	0.016709	357°525	35999°3720/cy	174°876	0°000	102°940
Mars	1.523692	0.093405	19°387	19140°3023/cy	49°557	1°849	336°059
Jupiter	5.202437	0.048402	250°327	3035°2275/cy	100°468	1°304	15°719
Saturn	9.551712	0.052340	267°246	1219°6465/cy	113°643	2°485	90°968
Uranus	19.293108	0.044846	118°432	424°8150/cy	74°090	0°773	176°615
Neptune	30.257162	0.007985	292°471	216°3047/cy	131°775	1°770	3°096
Pluto	39.783607	0.254351	8°230	143°4629/cy	110°386	17°120	224°742

5.2 Planetary Coordinates

To be able to calculate the acceleration at any given time using (5.6), we require the heliocentric coordinates $\mathbf{r}$ of the body concerned, and the corresponding coordinates $\mathbf{r}_i$ for the nine planets from Mercury to Pluto. For highly accurate calculations we may obtain the latter from previously calculated planetary ephemerides, such as those prepared by NASA's Jet Propulsion Laboratory and made available for scientific use. Although such ephemerides are highly accurate, they do, however, have the disadvantage of being extremely comprehensive (taking as many as 10 000 coefficients into account for each year). Luckily, in many cases, considerably simpler, approximate ephemerides are adequate for calculating the perturbations of a cometary or minor-planetary orbit to arc-second accuracy. Apart from close encounters, the perturbational acceleration is small in comparison with the acceleration produced by the Sun. As a result, small errors in the planetary coordinates normally produce negligible, second-order errors in the calculated coordinates of the perturbed body.

The simplest possibility is obviously that of making an approximation of the planetary coordinates oneself, using Keplerian orbits. The accuracy obtained in this way is, however, limited, especially for the outer planets. For predictions over a period of ten years, for example, the errors may amount to a few minutes of arc. An appropriate set of orbital elements for the period 1990 to 2000 is given in Table 5.2.

```

(*-----*)
(* POSITION: *)
(* Computes the position of a planet assuming Keplerian orbits. Mean *)
(* elements at epoch J2000 are used for Mercury to Mars; osculating *)
(* elements at epoch 1995/10/10 (JD 2450000.5) are used for Jupiter to *)
(* Pluto. Relative accuracy approx. 0.001 between 1990 and 2000. *)
(* *)
(* PLANET Name of the planet *)
(* T Time in Julian centuries since J2000 *)
(* X,Y,Z Ecliptic coordinates (equinox J2000) *)
(*-----*)

```

```

PROCEDURE POSITION ( PLANET: PLANET_TYPE; T: REAL; VAR X,Y,Z: REAL );

```

```

VAR  A,E,M,O,I,W,N : REAL;
      XX,YY,VVX,VVY : REAL;
      TO             : REAL;
      PQR            : REAL33;

```

```

BEGIN

```

```

  (* Heliocentric ecliptic orbital elements for equinox J2000 *)

```

```

  CASE PLANET OF

```

```

    MERCURY : BEGIN

```

```

      A:= 0.387099; E:=0.205634; M:=174.7947; N:=149472.6738;
      O:= 48.331;   I:= 7.0048;  W:= 77.4562;
    END;

```

```

    VENUS   : BEGIN

```

```

      A:= 0.723332; E:=0.006773; M:= 50.4071; N:= 58517.8149;
      O:= 76.680;   I:= 3.3946;  W:=131.5718;
    END;

```

```

    EARTH   : BEGIN

```

```

      A:= 1.000000; E:=0.016709; M:=357.5258; N:= 35999.3720;
      O:=174.876;   I:= 0.0000;  W:=102.9400;
    ENO;

```

```

    MARS     : BEGIN

```

```

      A:= 1.523692; E:=0.093405; M:= 19.3879; N:= 19140.3023;
      O:=49.557;    I:= 1.8496;  W:=336.0590;
    END;

```

```

    JUPITER : BEGIN

```

```

      A:= 5.202437; E:=0.048402; M:=250.3274; N:= 3035.2275;
      O:=100.4683; I:=1.3047;  W:= 15.7192;
    END;

```

```

    SATURN  : BEGIN

```

```

      A:= 9.551712; E:=0.052340; M:=267.2465; N:= 1219.6465;
      O:=113.6439; I:=2.4855;  W:= 90.9682;
    END;

```

```

    URANUS  : BEGIN

```

```

      A:=19.293108; E:=0.044846; M:=118.4320; N:= 424.8150;
      O:= 74.0903; I:=0.7733;  W:=176.6152;
    END;

```

NEPTUNE : BEGIN

A:=30.257162; E:=0.007985; M:=292.4716; N:= 216.3047;

O:=131.7750; I:=1.7700; W:= 3.0962;

END;

PLUTO : BEGIN

A:=39.783607; E:=0.254351; M:= 8.2304; N:= 143.4629;

O:=110.3865; I:=17.1201; W:=224.7424;

END;

END;

CASE PLANET OF

MERCURY, VENUS, EARTH, MARS : T0:= 0.0;

JUPITER, SATURN, URANUS, NEPTUNE, PLUTO : T0:=-0.042286105;

END;

M := M + N*(T-T0);

(* Cartesian coordinates mean ecliptic and equinox J2000 *)

GAUSVEC (O,I,W-0, PQR);

ELLIP (M,A,E, XX,YY,VVX,VVY);

ORBECL (XX,YY,PQR, X,Y,Z);

END;

(*-----*)

5.3 Numerical Integration

To simplify further discussion, the position $\mathbf{r}$ and velocity $\mathbf{v}$ of the body concerned may be combined as a single, six-dimensional vector

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix}, \quad (5.7)$$

known as the state vector. The change $\dot{\mathbf{y}}$ of position and velocity with time may therefore be expressed as a differential equation of the form

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}), \quad (5.8)$$

where the

$$\mathbf{f}(t, \mathbf{y}) = \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{a}(t, \mathbf{r}) \end{pmatrix} \quad (5.9)$$

function combines the velocity and acceleration, which are themselves dependent on time and position.

The basic, numerical integration procedure may be simply expressed using this convention. If we know the state vector $\mathbf{y}_0$ at a time t_0 , then we can calculate the approximate state vector at a later time $t_1 = t_0 + h$ from the approximation

$$\mathbf{y}(t_1) \approx \mathbf{y}_1 = \mathbf{y}(t_0) + h \cdot \dot{\mathbf{y}}(t_0) = \mathbf{y}_0 + h \cdot \mathbf{f}(t_0, \mathbf{y}_0) \quad (5.10)$$

If this step is carried out, we can then obtain an approximation for the state vector at time $t_2 = t_0 + 2h$ from t_1 and $\mathbf{y}_1$, and, in a similar way, corresponding approximate values for the position and velocity at any other times $t_i = t_0 + ih$. This simple procedure is not, however, particularly suitable for practical use, because it gives accurate results only when the step values h are very small, which means that an extremely large number of steps must be taken to calculate a complete orbit.

To obtain usable results with larger step values, the procedure adopted is the so-called multistep method, with improved approximations, which are based on several earlier values of $\mathbf{f}$. To illustrate the principle behind this procedure, let us assume that we have approximations $\mathbf{y}_j$ for the state vector $\mathbf{y}(t_j)$ at times $t_j = t_0 + jh$, where $j = 0, 1, \dots, i$. If we then integrate both sides of

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}) \quad (5.11)$$

with respect to time t from t_i to t_{i+1} , we then obtain the equivalent equation

$$\mathbf{y}(t_{i+1}) = \mathbf{y}(t_i) + \int_{t_i}^{t_i+h} \mathbf{f}(t, \mathbf{y}(t)) dt \quad (5.12)$$

The integral cannot be calculated directly in this form, however, because it depends on the unknown state vector $\mathbf{y}(t)$. To avoid this problem, the integrand is replaced by a polynomial $\mathbf{p}(t)$, that interpolates some of the values of the function

$$\mathbf{f}_j = \mathbf{f}(t_j, \mathbf{y}_j) \quad (5.13)$$

at earlier times t_j , which, in accordance with the assumption that we made above, are already known.

As an example, Fig. 5.2 shows a 3rd-order polynomial, which is determined by the four values $\mathbf{f}_{i-3}$, $\mathbf{f}_{i-2}$, $\mathbf{f}_{i-1}$ and $\mathbf{f}_i$ at times t_{i-3} , t_{i-2} , t_{i-1} and t_i . If this polynomial is written in the form

$$\mathbf{p}(t) = \mathbf{a}_0 + \mathbf{a}_1\sigma + \mathbf{a}_2\sigma^2 + \mathbf{a}_3\sigma^3 \quad \text{with} \quad \sigma = \frac{1}{h}(t - t_i) \quad (5.14)$$

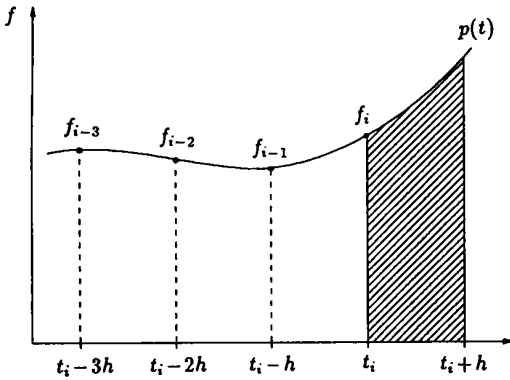


Fig. 5.2. Interpolating earlier values of a function and integrating for the next time interval

and the appropriate coefficients

$$\begin{aligned}
 a_0 &= (6f_i)/6 \\
 a_1 &= (-2f_{i-3} + 9f_{i-2} - 18f_{i-1} + 11f_i)/6 \\
 a_2 &= (-3f_{i-3} + 12f_{i-2} - 15f_{i-1} + 6f_i)/6 \\
 a_3 &= (-1f_{i-3} + 3f_{i-2} - 3f_{i-1} + 1f_i)/6
 \end{aligned} \tag{5.15}$$

are entered, we obtain the Adams-Bashforth 4th-order equation

$$y_{i+1} = y_i + \int_{t_i}^{t_i+h} p(t)dt = y_i + \frac{h}{24}(-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i), \tag{5.16}$$

which gives a good approximation of the state vector for larger step sizes. Repeating the procedure enables us to obtain corresponding values for subsequent times $t_i + jh$.

In general, the more terms are used in the interpolation polynomial (and thus the higher the degree), the higher the procedure's accuracy. With large step values, however, instabilities may occur, so the order of the polynomial and the step value must be appropriately chosen depending of the desired accuracy.

The DE subroutine, which will be used in what follows to calculate perturbed minor-planetary orbits, is also, in principle, an Adams multistep procedure. In contrast to the simplified form that has just been outlined - i.e., a multistep procedure with fixed order and step-width - DE has some special features that are of particular value for practical applications:

- The number of earlier points, and thus the order of the procedure is not fixed, but is independently selected by DE itself. When extremely high accuracies are required, DE uses 12th-order polynomials.
- Similarly, the step-width employed may vary from one step to the next, and thus be chosen to agree with the changes in the state vector with

time. This is particularly useful when dealing with highly eccentric orbits, which require significantly smaller steps near perihelion than at aphelion.

- In contrast to simple fixed-order, fixed-step-size, multistep procedures, no special initial calculation is required to obtain the first points $f_0, f_1, \dots$. Instead, DE begins with a small step of order 1, and increases the order and step-width at each step until optimal values are reached. Consequently, DE is self-starting.
- The state vector $y(t)$ may be obtained at arbitrary output times, and the values at the reference points used internally will be automatically interpolated. The choice of the step-size is not, therefore, affected by the choice of output points.

The DE procedure was developed by L.F. Shampine and M.K. Gordon, and implemented as a Fortran program. Because of space considerations, we can discuss neither the extremely comprehensive program code, nor details of the implementation. The reader is referred to the book *Computer Solution of Ordinary Differential Equations*, in which the basis of the procedure is thoroughly explained.

For our current application, DE has been translated into Pascal, and adapted to certain specific features of the latter language. Overall, the code includes the three sub-routines DE, STEP and INTRP, but where only DE is directly called by the user. The other two routines are used within DE, and are employed to calculate an individual integration step (STEP), and to interpolate the state vector at the initial points in time. For the Pascal version used here, certain constants and data types need to be globally declared:

```
CONST DE_EQN = 6;           (* Dimension of the state vector *)
      TWOU   = 4.0E-12;     (* Accuracy of the REAL data type *)
      FOURU  = 8.0E-12;

TYPE  DE_EQN_VECTOR    = ARRAY[1..DE_EQN] OF REAL;
      DE_12_VECTOR     = ARRAY[1..12] OF REAL;
      DE_13_VECTOR     = ARRAY[1..13] OF REAL;
      DE_PHI_ARRAY     = ARRAY[1..DE_EQN,1..16] OF REAL;
      DE_WORKSPACE_RECORD = RECORD
          YY,WT,P,YP,YPOUT    : DE_EQN_VECTOR;
          PHI                  : DE_PHI_ARRAY;
          ALPHA,BETA,V,W,PSI  : DE_12_VECTOR;
          SIG,G                : DE_13_VECTOR;
          X,H,HOLD,TOLD,DELSGN : REAL;
          NS,K,KOLD,ISNOLD    : INTEGER;
          PHASE1,START,NORND  : BOOLEAN;
      END;
```

The constant DE_EQN is the length of the state vector and serves to define the number of dimensions of various fields. When integrating individual minor-planet orbits, the state vector consists of precisely six components

(defining both position and velocity). In other applications, DE_EQN needs to be adjusted appropriately. The constants TWOU and FOURU are, respectively, twice and four times the value of the machine accuracy u . Depending on the computer and compiler, typical values of u for the REAL data type lie between 10^{-7} and 10^{-15} . For current PC compilers with 6-byte arithmetic $u \approx 2 \cdot 10^{-12}$.

The data type DE_EQN_VECTOR is a vector, and thus compatible with DE_EQN. It is used to store and transfer the state vector and related values with similar dimensions. The next three data types, DE_12_VECTOR, DE_13_VECTOR and DE_PHI_ARRAY, are mainly required for the declaration of certain, internal, intermediate values. Finally, DE_WORKSPACE_RECORD combines all those values in a single, global variable, which have to be stored between successive calls to DE.

```
(*-----*)
(* DE: variable order and step size multistep method of Shampine and *)
(*   Gordon. *)
(* F      Function to be integrated *)
(* NEQN   Number of differential equations *)
(* Y      State vector *)
(* T      Time *)
(* TOUT   Output time *)
(* RELERR Relative tolerance *)
(* ABSERR Absolute tolerance *)
(* IFLAG  Return code *)
(* WORK   Work area for global storage between subsequent calls *)
(*-----*)
PROCEDURE DE ( PROCEDURE F(X:REAL;Y:DE_EQN_VECTOR;VAR YP:DE_EQN_VECTOR);
              NEQN: INTEGER; VAR Y: DE_EQN_VECTOR; VAR T,TOUT: REAL;
              VAR RELERR,ABSERR: REAL; VAR IFLAG: INTEGER;
              VAR WORK: DE_WORKSPACE_RECORD );
```

The first parameter required by routine DE is the name of a subroutine of the form

```
PROCEDURE F( X:REAL; Y:DE_EQN_VECTOR; VAR YP:DE_EQN_VECTOR );
```

which calculates the derivative $YP = dy/dx$ of a differential equation $dy/dx = f(x,y)$ for given arguments $X = x$ und $Y = y$. A corresponding subroutine for the equation of motion of a minor planet, using the expressions for the acceleration as a function of position and time (which have been described earlier), may be written as follows:

```
(*-----*)
(* F: computes the time derivative of the state vector *)
(* X   Time (Modified Julian Date) *)
(* Y   State vector (x,y,z in AU, vx,vy,vz in AU/d) *)
(* YP  Derivative (vx,vy,vz in AU/d, ax,ay,az in AU/d**2) *)
(*-----*)
PROCEDURE F ( X: REAL; Y: DE_EQN_VECTOR; VAR DYDX: DE_EQN_VECTOR );
```

```

VAR T : REAL;
BEGIN
  (* Time in Julian centuries since J2000 *)
  T := ( X-51544.5 ) / 36525.0;
  (* Derivative of the state vector *)
  DYDX[1]:=Y[4]; DYDX[2]:=Y[5]; DYDX[3]:=Y[6];
  ACCEL ( T, Y[1],Y[2],Y[3], DYDX[4],DYDX[5],DYDX[6] );
END;
(*-----*)

```

The other parameters in DE are the number of differential equations (NEQN), the state vector (Y), the time (T), the output time (TOUT), the relative and the absolute accuracies required (RELERR, ABSERR), a status variable (IFLAG), and, finally, the workspace (WORK), which, as mentioned, is used to store internal data between calls to DE, and which therefore needs to be declared in the main program.

The first time DE is called, the starting time and the corresponding state vector are given by T and Y. The desired output time TOUT and the required relative and absolute accuracies RELERR and ABSERR also have to be entered. The status variable IFLAG is set to 1 to indicate to the integrator that the integration of a new problem is to be started.

Under normal circumstances, Y eventually contains the desired state vector at time TOUT. Simultaneously, T is assigned the value of TOUT, and the status variable IFLAG is set to 2 (= successful step). In all subsequent steps, only TOUT is assigned a new value by the user, while all the other variables (in particular, IFLAG) remain unaltered. Only at the beginning of a new problem or if the direction of integration is reversed does IFLAG have to be reset to one.

If, following a call to DE, IFLAG returns any value other than 2, the significance is as follows:

- | | |
|---------|---|
| IFLAG=3 | TOUT was not been reached, because either the required tolerance RELERR or ABSERR was too small. Both values should be appropriately increased for subsequent calculations. |
| IFLAG=4 | TOUT was not reached, because more than MAXNUM=500 steps were required internally. |
| IFLAG=5 | TOUT was not reached, because the differential equation appears to be too difficult to solve. With problems of celestial mechanics, this will probably not arise. |
| IFLAG=6 | Inadmissible input parameter (e.g., T=TOUT). |

In all cases, except when IFLAG=6, the integration may be directly resumed by calling DE again without altering any parameters. The interruption mainly serves to alert the user to any problems and to avoid (for example) non-terminating integrations.

5.4 Osculating Elements

For an unperturbed Keplerian orbit, the motion of a body around the Sun is specified by six orbital elements: semi-major axis a , eccentricity e , inclination i , longitude of the ascending node Ω , longitude of perihelion ω , and mean anomaly M . The orbital elements are therefore constants, from which the position and orbital velocity may be calculated at any required time (cf. Chap. 4).

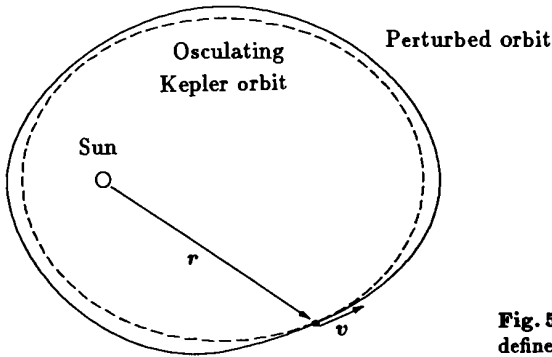


Fig. 5.3. Using osculating elements to define a perturbed orbit

Because of their geometrical significance, these orbital elements are particularly easy to understand, so it is advisable to use them to describe a perturbed minor-planet or cometary orbit. We are able to utilize the fact that the orbital elements and the state vector are linked in an unambiguous way. If, at a specific time t we have the position $\mathbf{r} = (x, y, z)$ and velocity $\dot{\mathbf{r}} = (\dot{x}, \dot{y}, \dot{z})$ of a celestial body, then there is a corresponding set of specific orbital elements $(a, e, i, \Omega, \omega, M)$, from which we can obtain the state vector $(\mathbf{r}, \mathbf{v})$, using the unperturbed Kepler equations. For perturbed motion, orbital elements may be defined in a similar way. These are no longer constant, however, but exhibit various fluctuations with time depending on the form and degree of perturbation. These orbital elements derived from the state vector are normally described as *osculating* elements, because the corresponding Kepler orbit 'kisses' the true orbit at the appropriate epoch, and is very close to it for a certain period of time (Fig. 5.3).

As for unperturbed motion, the osculating orbital plane is defined at any point in time by the position and velocity vector (see also Fig. 4.1). The cross product of $\mathbf{r}(t)$ and $\dot{\mathbf{r}}(t)$:

$$\mathbf{h} = \mathbf{r} \times \dot{\mathbf{r}} = \begin{pmatrix} y\dot{z} - z\dot{y} \\ z\dot{x} - x\dot{z} \\ x\dot{y} - y\dot{x} \end{pmatrix} \quad (5.17)$$

therefore defines a vector $h(t)$, that is perpendicular to the corresponding orbital plane, and is therefore parallel to the Gaussian vector normal to the orbit:

$$\mathbf{R} = \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} +\sin i \sin \Omega \\ -\sin i \cos \Omega \\ +\cos i \end{pmatrix} = \begin{pmatrix} h_x/h \\ h_y/h \\ h_z/h \end{pmatrix} . \quad (5.18)$$

From this we may obtain the osculating orbital inclination i and the longitude of the ascending node Ω as

$$i = \tan^{-1} \left(\frac{\sqrt{h_x^2 + h_y^2}}{h_z} \right) \quad \Omega = \tan^{-1} \left(\frac{h_x}{-h_y} \right) . \quad (5.19)$$

By solving (4.20) for u we also obtain the additional expression

$$u = \nu + \omega = \tan^{-1} \left(\frac{zh}{-xh_y + yh_x} \right) \quad (5.20)$$

as the argument for the latitude u , which we shall require later to calculate the argument of perihelion.

As may be seen by the relevant equations (4.5) and (4.10), the semi-major axis of the orbit depends solely on the distance r from the Sun, and the magnitude $v = |\dot{\mathbf{r}}|$ of the velocity:

$$a = \left(\frac{2}{r} - \frac{v^2}{GM_\odot} \right)^{-1} . \quad (5.21)$$

Accordingly, we always obtain an ellipse with a positive semi-major axis as an osculating Kepler orbit if the velocity of the body relative to the Sun is smaller than the corresponding escape velocity $v_\infty = \sqrt{2GM_\odot/r}$.

In this case we may rewrite the equations (4.5) and (4.10) for position and velocity as functions of the eccentric anomaly, to give us

$$\begin{aligned} e \cos(E) &= 1 - r/a \\ e \sin(E) &= (x\dot{x} + y\dot{y} + z\dot{z})/\sqrt{GM_\odot a} \end{aligned} \quad (5.22)$$

and enable us to determine the eccentricity e and the eccentric anomaly E . From the Kepler equation (4.7) we then obtain the required mean anomaly

$$M = E - \frac{180^\circ}{\pi} e \sin(E) \quad (5.23)$$

at time t . Finally, from the true anomaly

$$\nu = \tan^{-1} \frac{\sqrt{1 - e^2} \sin(E)}{\cos(E) - e} \quad (5.24)$$

corresponding to E , we obtain the sixth orbital element, the argument of perihelion

$$\omega = \nu - u \quad . \quad (5.25)$$

There are corresponding expressions for parabolic and hyperbolic orbits, which are, however, not required for calculation of the orbits of minor planets or periodic comets. The following XYZKEP subroutine is therefore restricted to elliptical orbits alone.

```
(*-----*)
(*)
(*) XYZKEP: conversion of the state vector into Keplerian elements (*)
(*) for elliptical orbits (*)
(*)
(*) X,Y,Z : Position [AU] (*)
(*) VX,VY,VZ : Velocity [AU/d] (*)
(*) AX : Semi-major axis [AU] (*)
(*) ECC : Eccentricity (*)
(*) INC : Inclination [deg] (*)
(*) LAN : Longitude of the ascending node [deg] (*)
(*) AOP : Argument of perihelion [deg] (*)
(*) MA : Mean anomaly [deg] (*)
(*)
(*)-----*)
```

```
PROCEDURE XYZKEP ( X,Y,Z, VX,VY,VZ : REAL;
                  VAR AX,ECC,INC,LAN,AOP,MA : REAL );
```

```
CONST DEG = 57.29577951308; (* Conversion from radian to degrees *)
KGAUSS = 0.01720209895; (* Gaussian gravitational constant *)
```

```
VAR HX,HY,HZ,H, R,V2 : REAL;
    GM, C,S,E2, EA,U,NU : REAL;
```

```
BEGIN
```

```
HX := Y*VZ-Z*VY; (* Areal velocity *)
HY := Z*VX-X*VZ;
HZ := X*VY-Y*VX;
H := SQRT ( HX*HX + HY*HY + HZ*HZ );
```

```
LAN := ATN2 ( HX, -HY ); (* Long. ascend. node *)
INC := ATN2 ( SQRT(HX*HX+HY*HY), HZ ); (* Inclination *)
U := ATN2 ( Z*H, -X*HY+Y*HX ); (* Arg. of latitude *)
```

```
GM := KGAUSS*KGAUSS;
R := SQRT ( X*X + Y*Y + Z*Z ); (* Distance *)
V2 := VX*VX + VY*VY + VZ*VZ; (* Velocity squared *)
```

```
AX := 1.0 / (2.0/R-V2/GM); (* Semi-major axis *)
```

```
C := 1.0-R/AX; (* e*cos(E) *)
```

```

S  := (X*VX+Y*VY+Z*VZ)/(SQRT(AX)*KGAUSS);      (* e*sin(E)          *)

E2  := C*C + S*S;
ECC := SQRT ( E2 );                             (* Eccentricity      *)
EA  := ATN2 ( S, C );                             (* Eccentric anomaly *)

MA  := EA - S*DEG;                                (* Mean anomaly      *)

NU  := ATN2 ( SQRT(1.0-E2)*S, C-E2 );             (* True anomaly      *)
AOP := U - NU;                                    (* Arg. of perihelion *)

IF (LAN<0.0) THEN LAN:=LAN+360.0;
IF (AOP<0.0) THEN AOP:=AOP+360.0;
IF (MA <0.0) THEN MA :=MA +360.0;

END;
(*-----*)

```

5.5 The NUMINT Program

The NUMINT program may be used for accurate calculation of the orbits of minor planets or periodic comets. Its structure and use closely resemble the COMET program discussed in Chap. 4. Through the use of a numerical integration procedure both the Sun's gravitational attraction and perturbations caused by the major planets may be taken into account in orbital predictions. In particular, significantly more accurate results may be obtained for predictions covering long periods of time. Against this higher accuracy there is the disadvantage that the time required for the integration increases in proportion to the span of the ephemerides required, and is generally significantly greater than for the COMET program.

When the program is called, the orbital elements and the corresponding equinox are read from the data file NUMINT.DAT and displayed on the screen for checking purposes, before the user enters the time-span, the step value, and the equinox required for the ephemerides. Using these values, the position and velocity of the minor planet are determined relative to the fixed equinox of J2000, and then predicted for the beginning of the ephemerides by numerical integration from the epoch of the orbital elements. If one is working with orbital elements whose epoch is some years earlier than the starting date, then the integration may take a number of minutes, depending on the computer's processing power.

Using the state vector thus obtained, NUMINT then calculates the corresponding osculating elements and displays these on the screen. From the changes in the individual elements relative to those for the original epoch one can gain a good idea of the effect of the perturbations caused by the major planets. In addition, the orbital elements thus obtained may be used, when repeating calculations with old orbital elements, to avoid having to span major intervals of time prior to the start of the ephemerides.

Finally, as in the COMET program, an ephemeris of the minor planet is calculated, using the required step-width and duration, and output. Light-time is taken into account, so the resulting, astrometric coordinates may be directly compared with star charts or catalogues. The position of the Earth relative to the Sun is calculated – again as in COMET – using the SUN200 subroutine, to avoid errors in the conversion of heliocentric to geocentric coordinates for the minor planet.

```
(*-----*)
(*)
(*)          NUMINT          (*)
(*)      Numerical integration of perturbed minor planet orbits (*)
(*)              version 93/07/01 (*)
(*)
(*)-----*)
```

```
PROGRAM NUMINT ( INPUT,OUTPUT, NUMINP );
```

```
CONST J2000 = 0.0;
      DE_EQN = 6;          (* Dimension of the state vector *)
      TWOU = 4.0E-12;      (* Accuracy of the Turbo Pascal REAL *)
      FOURU = 8.0E-12;     (* data type u=2**(-39)=1.81E-12 *)
```

```
TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
      PLANET_TYPE = ( MERCURY, VENUS, EARTH, MARS,
                      JUPITER, SATURN, URANUS, NEPTUNE, PLUTO );
      DE_EQN_VECTOR = ARRAY[1..DE_EQN] OF REAL;
      DE_12_VECTOR  = ARRAY[1..12] OF REAL;
      DE_13_VECTOR  = ARRAY[1..13] OF REAL;
      OE_PHI_ARRAY  = ARRAY[1..DE_EQN,1..16] OF REAL;
      DE_WORKSPACE_RECORD = RECORD
                                YY,WT,P,YP,YPOUT : DE_EQN_VECTOR;
                                PHI                : DE_PHI_ARRAY;
                                ALPHA,BETA,V,W,PSI : DE_12_VECTOR;
                                SIG,G              : OE_13_VECTOR;
                                X,H,HOLD,TOLD,DELSGN : REAL;
                                NS,K,KOLD,ISNOLD   : INTEGER;
                                PHASE1,START,NORND : BOOLEAN;
      END;
```

```
VAR DAY,MONTH,YEAR,NLINE,IFLAG : INTEGER;
      D,HOUR,T_EPOCH, TEQX0,TEQX, FAC : REAL;
      A,E,M,INC,LAN,AOP : REAL;
      MJD_START,MJD_END, T,T1,DT,T2 : REAL;
      XX,YY,ZZ, VX,VY,VZ, XS,YS,ZS : REAL;
      L,B,R,LS,BS,RS,RA,DEC,DELTA,DELTA0 : REAL;
      PQR : REAL33;
      EQX0_TO_J2000, J2000_TO_EQX, AS : REAL33;
      Y : DE_EQN_VECTOR;
      WORK : DE_WORKSPACE_RECORD;
      NUMINP : TEXT;
```

```
(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN,CS,TN,ACS,ATN,ATN2,POLAR,CART,DMS,CUBR, MJD,CALDAT, *)
(* ECLEQU,GAUSVEC,ORBECL, PMATECL,PRECART, ECCANOM,ELLIP,XYZKEP, *)
(* SUN200, POSITION, INTRP,STEP,DE, ACCEL, F *)
(*-----*)
```

```
(*-----*)
(* WRTELM: writes orbital elements *)
(*-----*)
```

```
PROCEDURE WRTELM ( YEAR,MONTH: INTEGER; D: REAL;
                  A,E,INC,LAN,AOP,M, TEQX: REAL );
```

```
BEGIN
```

```
WRITELN(' Epoch (y m d)           ',YEAR:7,MONTH:3,D:6:2);
WRITELN(' Semi-major axis (a)       ', A:14:7,' AU ');
WRITELN(' Eccentricity (e)           ', E:14:7);
WRITELN(' Inclination (i)              ',INC:12:5,' deg');
WRITELN(' Long. of ascending node ',LAN:12:5,' deg');
WRITELN(' Argument of perihelion ',AOP:12:5,' deg');
WRITELN(' Mean anomaly (M)           ', M:12:5,' deg');
WRITELN(' Equinox                      ',2000.0+100.0*TEQX:9:2);
```

```
END;
```

```
(*-----*)
(* GETELM: reads orbital elements from file NUMINP *)
(*-----*)
```

```
PROCEDURE GETELM (VAR T,A,E,M: REAL; VAR PQR: REAL33; VAR TEQX: REAL);
```

```
VAR YEAR,MONTH,DAY : INTEGER;
    D,HOUR          : REAL;
    INC,LAN,AOP     : REAL;
```

```
BEGIN
```

```
WRITELN;
WRITELN(' NUMINT ');
WRITELN(' Numerical integration of perturbed minor planet orbits ');
WRITELN(' version 93/07/01 ');
WRITELN(' (c) 1993 Thomas Pfleger, Oliver Montenbruck ');
WRITELN;
```

```
WRITELN(' Orbital elements from file NUMINP: ');
```

```
WRITELN;
```

```
(* Open file for reading *)
```

```
RESET(NUMINP); (* Standard Pascal *)
```

```
(* ASSIGN(NUMINP,'NUMINP.DAT'); RESET(NUMINP); *) (* TURBO Pascal *)
```

```
(* Display orbital elements *)
```

```
READLN (NUMINP,YEAR,MONTH,D);
```

```
READLN (NUMINP,A);
```

```
READLN (NUMINP,E);
```

```
READLN (NUMINP,INC);
```

```
READLN (NUMINP,LAN);
```

```
READLN (NUMINP,AOP);
```

```
READLN (NUMINP,M);
```

```
READLN (NUMINP,TEQX);
```

```
DAY:=TRUNC(D); HOUR:=24.0*(D-DAY);
```

```
T := ( MJD(DAY,MONTH,YEAR,HOUR) - 51544.5) / 36525.0;
```

```

      TEQX0 := (TEQX0-2000.0)/100.0;
      GAUSVEC(LAN,INC,AOP,PQR);
END;

(*-----*)
(* GETEPH: reads desired period of time and equinox of the ephemeris *)
(*-----*)
PROCEDURE GETEPH ( VAR T1,DT,T2,TEQX: REAL );
  VAR YEAR,MONTH,DAY : INTEGER;
      EQX,HOURL,JD : REAL;
BEGIN
  Writeln;
  Writeln(' Begin and end of the ephemeris: ');
  Writeln;
  WRITE ( ' First date (yyyy mm dd hh.hhh)           ... ');
  READLN (YEAR,MONTH,DAY,HOURL);
  T1 := ( MJD(DAY,MONTH,YEAR,HOURL) - 51544.5 ) / 36525.0;
  WRITE ( ' Final date (yyyy mm dd hh.hhh)           ... ');
  READLN (YEAR,MONTH,DAY,HOURL);
  T2 := ( MJD(DAY,MONTH,YEAR,HOURL) - 51544.5 ) / 36525.0;
  WRITE ( ' Step size (dd hh.hh)                     ... ');
  READLN (DAY,HOURL);
  DT := ( DAY + HOURL/24.0 ) / 36525.0;
  Writeln;
  WRITE ( ' Desired equinox of the ephemeris (yyyy.y) ... ');
  READLN (EQX);
  TEQX := (EQX-2000.0)/100.0;
END;

(*-----*)
(* WRTLBR: write L and B in deg,min,sec and R *)
(*-----*)
PROCEDURE WRTLBR(L,B,R:REAL);
  VAR H,M: INTEGER;
      S : REAL;
BEGIN
  DMS(L,H,M,S); WRITE (H:5,M:3,S:5:1);
  DMS(B,H,M,S); Writeln(H:5,M:3,TRUNC(S+0.5):3,R:11:6);
END;

(*-----*)
(* INTEGRATE: Integrates the equation of motion *)
(* Y State vector (x,y,z in AU, vx,vy,vz in AU/d) *)
(* MJD Epoch (Modified Julian Date) *)
(* MJD_END Final epoch (Modified Julian Date) *)
(* IFLAG Return code *)
(* WORK Work space *)
(*-----*)
PROCEDURE INTEGRATE ( VAR Y : DE_EQN_VECTOR;
                      VAR MJD, MJD_END : REAL;
                      VAR IFLAG : INTEGER;
                      VAR WORK : DE_WORKSPACE_RECORD );
  CONST EPS = 1.0E-8; (* Accuracy *)

```

```

VAR RELERR, ABSERR: REAL;
BEGIN
  RELERR := EPS;      (* Relative accuracy requirement *)
  ABSERR := 0.0;      (* Absolute accuracy requirement *)
  IF ( MJD_END <> MJD ) THEN
    BEGIN
      REPEAT
        DE ( F, 6, Y, MJD, MJD_END, RELERR,ABSERR, IFLAG, WORK );
      UNTIL ((ABS(IFLAG)=2) OR (IFLAG=6));
      IF (IFLAG=6) THEN WRITELN ( ' Illegal input in DE ');
    END;
  END;

  (*-----*)

  BEGIN (* NUMINT *)

    (* Read orbital elements, prediction interval and equinox *)

    GETELM (T_EPOCH,A,E,M,PQR,TEQX0);
    GETEPH (T1,DT,T2,TEQX);

    (* Calculate precession matrices *)

    PMATECL (TEQX0,J2000,EQX0_TO_J2000);
    PMATECL (J2000,TEQX,J2000_TO_EQX);

    (* Heliocentric position and velocity vector at epoch *)
    (* referred to ecliptic and equinox of J2000 *)

    ELLIP ( M,A,E, XX,YY,VX,VY );
    ORBECL ( XX,YY,PQR, Y[1],Y[2],Y[3] );
    ORBECL ( VX,VY,PQR, Y[4],Y[5],Y[6] );
    PRECART ( EQX0_TO_J2000,Y[1],Y[2],Y[3] );
    PRECART ( EQX0_TO_J2000,Y[4],Y[5],Y[6] );

    (* Start integration: propagate state vector from epoch *)
    (* to start of ephemeris *)

    MJD_START := T_EPOCH*36525.0 + 51544.5;
    MJD_END   := T1*36525.0 + 51544.5;
    IFLAG     := 1;  (* Initialization flag *)

    INTEGRATE ( Y, MJD_START, MJD_END, IFLAG, WORK );

    (* Orbital elements at start of ephemeris *)

    XX:=Y[1]; YY:=Y[2]; ZZ:=Y[3];      (* Copy J2000 state vector *)
    VX:=Y[4]; VY:=Y[5]; VZ:=Y[6];

    PRECART (J2000_TO_EQX,XX,YY,ZZ);      (* Precession *)
    PRECART (J2000_TO_EQX,VX,VY,VZ);

```



```

XYZKEP ( XX,YY,ZZ,VX,VY,VZ, A,E,INC,LAN,AOP,M ); (* Convert *)

WRITELN; WRITELN;
WRITELN ( ' Orbital elements at start epoch:' );
WRITELN;

CALDAT ( MJD_END, DAY, MONTH, YEAR, HOUR );

WRITELN ( YEAR, MONTH, DAY+HOUR/24.0,          (* Print elements *)
          A,E,INC,LAN,AOP,M, TEQX );

(* Create ephemeris *)

WRITELN; WRITELN;
WRITELN ( '      Date      ET   Sun    l      b      r',
          '      RA      Dec    Distance ');
WRITELN ( ' ':45,' h m s      o '' "      (AU) ');

NLINE := 0;
T      := T1;

REPEAT

  (* Integrate orbit to time T *)

  MJD_END := T*36525.0 + 51544.5;
  INTEGRATE ( Y, MJO_START, MJD_END, IFLAG, WORK );

  (* Heliocentric ecliptic coordinates, equinox TEQX *)

  XX:=Y[1]; YY:=Y[2]; ZZ:=Y[3];      (* Copy J2000 state vector *)
  VX:=Y[4]; VY:=Y[5]; VZ:=Y[6];

  PRECART ( J2000_TO_EQX, XX, YY, ZZ );          (* Precession *)
  PRECART ( J2000_TO_EQX, VX, VY, VZ );

  POLAR ( XX, YY, ZZ, R, B, L );

  (* Ecliptic coordinates of the Sun, equinox TEXQ *)

  SUN200 ( T, LS, BS, RS );  CART ( RS, BS, LS, XS, YS, ZS );
  PMATECL ( T, TEQX, AS );  PRECART ( AS, XS, YS, ZS );

  (* Geometric geocentric coordinates *)

  XX:=XX+XS; YY:=YY+YS; ZZ:=ZZ+ZS;

  (* First order correction for light travel time *)

  DELTA0 := SQRT ( XX*XX + YY*YY + ZZ*ZZ );
  FAC      := 0.00578*DELTA0;

  XX:=XX-FAC*VX; YY:=YY-FAC*VY; ZZ:=ZZ-FAC*VZ;

```

```

ECLEQU (TEQX,XX,YY,ZZ);
POLAR (XX,YY,ZZ,DELTA,DEC,RA); RA:=RA/15.0;

(* Output *)

CALDAT (MJD_END,DAY,MONTH,YEAR, HOUR);
WRITE(YEAR:4,'/',MONTH:2,'/',DAY:2,HOUR:6:1);
WRITE(LS:7:1,L:7:1,B:6:1,R:7:3); WRTLBR(RA,DEC,DELTA0);
NLINE := NLINE+1; IF (NLINE MOD 5) = 0 THEN WRITELN;

(* Next time step *)

T := T + DT;

UNTIL (T2<T);

END.
(*-----*)

```

Table 5.3. Osculating orbital elements for minor planet Ceres

Epoche	1983 Sep. 23.0 ET	1992 Jun. 27.0 ET
<i>a</i>	2.7657991 AE	2.7674385 AE
<i>e</i>	0.0785650	0.0765551
<i>i</i> (1950)	10°60646	10°59950
(2000)	10°60695	10°59999
Ω (1950)	80°05225	80°01288
(2000)	80°71588	80°67649
ω (1950)	73°07274	71°07929
(2000)	73°10812	71°11469
<i>M</i>	174°19016	141°46349

As an example of how this program may be applied we will calculate an ephemeris for the minor planet 1, Ceres around the time of opposition in 1992, from osculating elements for the year 1983. The first line of the file NUMINT.DAT contains the epoch of the elements, followed by the six orbital elements $a, e, i, \Omega, \omega, M$ in sequence, and finally the equinox of the elements.

```

1983 09 23.0 ! Epoch (year month day.fraction) Minor planet 1 Ceres
2.7657991 ! Semi-major axis a in AU
0.0785650 ! Eccentricity e
10.60646 ! Inclination i (deg)
80.05225 ! Longitude of the ascending node (deg)
73.07274 ! Argument of perihelion (deg)
174.19016 ! Mean anomaly (deg)
1950.0 ! Equinox of the orbital elements

```

Corresponding data for other numbered minor planets are published yearly in the *Ephemerides of Minor Planets* (EMP) by the Institute for Theoretical Astronomy in St. Petersburg (formerly Leningrad), and in the Smithsonian Astrophysical Observatory's *Minor Planet Circulars*, as well as in various other yearbooks.

The method of entering the required time-span and the step-size for the ephemeris and the equinox is similar to that in the COMET program. As usual, the user's input is indicated in *italics*.

NUMINT

Numerical integration of perturbed minor planet orbits
version 93/07/01

(c) 1993 Thomas Pfleger, Oliver Montenbruck

Orbital elements from file NUMINP:

```
Epoch (y m d)          1983  9 23.00
Semi-major axis (a)      2.7657991 AU
Eccentricity (e)         0.0785650
Inclination (i)          10.60646 deg
Long. of ascending node  80.05225 deg
Argument of perihelion   73.07274 deg
Mean anomaly (M)         174.19016 deg
Equinox                  1950.00
```

Begin and end of the ephemeris:

```
First date (yyyy mm dd hh.hhh)    ... 1992 06 27 00.0
Final date (yyyy mm dd hh.hhh)    ... 1992 07 25 00.0
Step size (dd hh.hh)              ... 2 00.0
```

Desired equinox of the ephemeris (yyyy.y) ... 2000.0

Orbital elements at start epoch:

```
Epoch (y m d)          1992  6 27.00
Semi-major axis (a)      2.7674385 AU
Eccentricity (e)         0.0765551
Inclination (i)          10.59999 deg
Long. of ascending node  80.67649 deg
Argument of perihelion   71.11469 deg
Mean anomaly (M)         141.46349 deg
Equinox                  2000.00
```

Date	ET	Sun	l	b	r	RA h m s	Dec ° ' "	Distance (AU)
1992/ 6/27	0.0	95.6	297.9	-6.5	2.939	20 55 54.2	-27 0 46	2.042442
1992/ 6/29	0.0	97.5	298.2	-6.5	2.940	20 54 53.1	-27 13 47	2.029176
1992/ 7/ 1	0.0	99.4	298.6	-6.6	2.941	20 53 46.4	-27 26 57	2.016813
1992/ 7/ 3	0.0	101.3	299.0	-6.6	2.942	20 52 34.4	-27 40 13	2.005382
1992/ 7/ 5	0.0	103.2	299.4	-6.7	2.942	20 51 17.3	-27 53 33	1.994909

1992/ 7/ 7	0.0	105.1	299.7	-6.7	2.943	20 49 55.3	-28 6 53	1.985418
1992/ 7/ 9	0.0	107.0	300.1	-6.8	2.944	20 48 28.8	-26 20 11	1.976927
1992/ 7/11	0.0	108.9	300.5	-6.8	2.945	20 46 56.0	-26 33 24	1.969455
1992/ 7/13	0.0	110.9	300.9	-6.9	2.946	20 45 23.3	-28 46 29	1.963017
1992/ 7/15	0.0	112.8	301.2	-6.9	2.946	20 43 45.1	-28 59 23	1.957629
1992/ 7/17	0.0	114.7	301.6	-7.0	2.947	20 42 3.7	-29 12 3	1.953304
1992/ 7/19	0.0	116.6	302.0	-7.0	2.946	20 40 19.4	-29 24 27	1.950055
1992/ 7/21	0.0	118.5	302.4	-7.1	2.949	20 36 32.8	-29 36 32	1.947892
1992/ 7/23	0.0	120.4	302.7	-7.1	2.950	20 36 44.3	-29 48 14	1.946825
1992/ 7/25	0.0	122.3	303.1	-7.2	2.950	20 34 54.2	-29 59 33	1.946860

Following the data input, a few minutes – depending on the computer – are required for the orbit to be integrated over nine years, before the osculating elements for 1992 Jun.27 are displayed. Subsequent calculation of the ephemeris over a period of one month takes merely a few seconds.

Comparison with a yearbook shows that, despite the simplifications made in calculating the planetary coordinates, the accuracy of the ephemeris is about 1", and is therefore of the same degree of accuracy as the solar coordinates used in converting between heliocentric and geocentric coordinates.

If the calculation is carried out without the perturbations caused by the planets, such as would be possible, for example, by commenting out the appropriate lines in the ACCEL subroutine, then the following ephemeris is obtained:

Date	ET	Sun	l	b	r	RA			Dec			Distance (AU)
						h	m	s	o	'	"	
1992/ 6/27	0.0	95.6	298.6	-6.6	2.938	21	0	35.9	-26	49	55	2.049522
1992/ 6/29	0.0	97.5	299.0	-6.6	2.939	20	59	37.9	-27	2	56	2.035691
...												
1992/ 7/23	0.0	120.4	303.5	-7.2	2.950	20	41	55.4	-29	39	16	1.948277
1992/ 7/25	0.0	122.3	303.9	-7.3	2.951	20	40	6.3	-29	50	51	1.947831

It will be seen that the perturbations mainly affect the heliocentric longitude of the minor planet, and amount to 0°7–0°8, while the latitude shows smaller, but still significant differences amounting to about 0°1. Because of the smaller distance to the Earth, the right ascension of the unperturbed ephemeris differs from the actual orbit by about five minutes of time or 1°3 angular degrees.

6. Planetary Orbits

Excluding Mercury and Pluto, the major planets are very similar in the low eccentricities and low inclinations of their orbits relative to the ecliptic. Cometary orbits, by contrast, are distributed at random in space and have very varied shapes. The near-circular orbits of the planets are so widely separated that they never come close to one another.

These properties of planetary orbits have two important consequences for the calculation of ephemerides. The first concerns the way of handling the non-perturbed Kepler orbit, which was described in Chap. 4 for the various general forms of orbit. Because of the low eccentricity, the true anomaly shows only slight periodic departures from the mean anomaly. It can therefore be determined quicker by using a series expansion than by solving Kepler's equation. The same applies to the distance from the Sun, which in general differs only by a range of about one percent from the value of the semi-major axis. Furthermore, the orbit's departure from the ecliptic can be described by a brief approximation, thus doing away with the complication of calculating Gaussian vectors. These approximations will be described in greater detail in the next section.

The second point concerns the alterations to planetary orbits as a result of their mutual gravitational forces. Because close approaches of the planets do not occur, these forces are always several orders of magnitude smaller than the Sun's gravitational attraction. Major changes in orbits, like those that occur when comets approach Jupiter or Saturn, are not found among planetary motions. Their orbits may therefore be described by average Kepler ellipses, on which are superimposed small periodic perturbations.

In general, planetary ephemerides are nowadays calculated by using numerical integration methods, the results being stored in an appropriate form on magnetic media. Thanks to the introduction of modern computers, this method is remarkable for its simplicity and accuracy. There is a corresponding disadvantage, however, in that an analytical description of the planetary motions is no longer available, as was the case with the purely two-body problem. One of the greatest achievements of celestial mechanics was, and remains, the derivation of series expansions from which the deviations of planetary orbits from unperturbed motion can be calculated for any given time. As can be seen from the length of these series, their derivation is an exceptionally laborious and long-winded process, which is far beyond the scope of this book. For our purposes—the accurate calculation of planetary positions—however,

it suffices to have an understanding of the basic structure of the series expansions and how they may be calculated mathematically. Anyone taking a greater interest in the development of perturbation series should refer to the various textbooks on celestial mechanics.

Among the best-known analytical descriptions of planetary motions are the tables of the motions of the inner planets from Mercury to Mars, prepared by Simon Newcomb at the U.S. Naval Observatory. Although they were produced around the turn of the century, until only a few years ago they still formed the basis of the corresponding ephemerides in the *Astronomical Almanac*. There are similar tables by G.W. Hill for Jupiter and Saturn, and by Newcomb for Uranus and Neptune. These were soon replaced by the first numerically integrated ephemerides, however, because they did not attain the same accuracy over long periods of time as the tables for the inner planets.

The works mentioned, together with the series expansion for the orbit of Pluto by E. Goffin, J. Meeus and C. Steyart, form the basis for this chapter. The various formulae for the perturbation terms sometimes differ so greatly for individual planets, however, that it is not possible to convert them directly into a program in a simple and uniform manner. Because of this, all the series were first processed by a formula-manipulation program. This enabled the Kepler motion and the perturbation terms to be expressed independently of their original description as periodic motion in longitude, latitude, and radius. In addition, a large number of terms were excluded, which, because of their low values (generally $\ll 1''$), have no influence on the observed accuracy of the series. This reorganisation allows comparatively short and compact programming, without reducing the accuracy of the expansions to any significant degree.

6.1 Series Expansion of the Kepler Problem

The orbit of a planet in an unperturbed Kepler ellipse is a strictly periodic function of the mean anomaly M . If the mean anomaly alters by 360° , the planet has completed one orbit and reached exactly the same point. Any function of the orbital position can therefore be expressed as a Fourier series, i.e., as the sum of the sine and cosine functions for multiple arguments of M . Examples of this are the series expansions

$$\nu - M = a_1 \sin(M) + a_2 \sin(2M) + a_3 \sin(3M) + \dots$$

for the difference between true and mean anomaly and that for the distance r in units of the semi-major axis a :

$$r/a = b_0 + b_1 \cos(M) + b_2 \cos(2M) + \dots$$

The coefficients a_i and b_i in these series are functions of the orbital eccentricity e . The most important terms for $\nu - M$ (known as the *equation of the centre*)

and r/a up to powers of e^2 are:

$$\nu - M = 2e \sin(M) + \frac{5}{4}e^2 \sin(2M) + \dots \quad (\text{radians}) \quad (6.1)$$

and

$$r/a = \left(1 + \frac{1}{2}e^2\right) - e \cos(M) - \frac{1}{2}e^2 \cos(2M) + \dots \quad (6.2)$$

For small eccentricities both series converge very quickly, because from term to term the individual components become smaller by a factor that is of the order of e . For the Earth's orbit, where $e = 0.0167$, we have

$$\begin{aligned} \nu &= M + 0.0334 \sin(M) + 0.00035 \sin(2M) + \dots \quad (\text{radians}) \\ &= M + 1^\circ 9' 16'' \sin(M) + 0^\circ 0' 20'' \sin(2M) + \dots \\ r &= a \cdot (1.00014 - 0.0167 \cos(M) - 0.00014 \cos(2M) + \dots) \end{aligned}$$

The error in these shortened formulae amounts to only about 1 arc-second or 10^{-5} AU, and can be reduced even further if one includes additional terms.

Because the full description of these formulae requires a rather deeper understanding of the mathematics involved, we will only derive the first term of the equation of the centre. We begin by taking Kepler's equation (4.7):

$$E - e \sin E = M \quad .$$

As E and M differ by a term of order e only, we may substitute $e \sin M$ for $e \sin E$ to a close approximation, and obtain

$$E - M \approx e \sin M$$

for the difference between the eccentric and the true anomaly. In addition, from the equation of a conic section $r = a(1 - e^2)/(1 + e \cos \nu)$, where $r \cos \nu = a(\cos E - e)$, it follows that

$$\cos \nu \approx \cos E - e(1 - \cos \nu \cos E) \quad .$$

In the second term on the right-hand side we can now again substitute $\cos E$ for $\cos \nu$, obtaining

$$\cos \nu \approx \cos E - \{e \sin E\} \sin E \quad .$$

If we compare this with the Taylor expansion of the cosine function,

$$\cos(x + \Delta x) \approx \cos x - \Delta x \sin x \quad ,$$

then we can see that for small eccentricities, $\nu - E$ is approximately equal to $e \sin E$. Therefore, as indicated, the first component of the equation of the centre is

$$\nu - M = (\nu - E) + (E - M) \approx e \sin E + e \sin M \approx 2e \sin M \quad .$$

For small orbital inclinations i , the orbits are always close to the ecliptic. A planet's ecliptic longitude l therefore differs from the sum $\varpi + \nu$ of the perihelion longitude and the true anomaly only by a small correction R (known as the *reduction to the ecliptic*). R disappears for $i = 0$, and for small inclinations has, to a first approximation, a value of

$$R = l - (\nu + \varpi) \approx -\tan^2\left(\frac{i}{2}\right) \cdot \sin(2u) \quad (\text{radians}) \quad (6.3)$$

where $u = \nu + \omega = \nu + \varpi - \Omega$. As $\nu \approx M$ ($u \approx M + \omega$), R may also be expressed in the form

$$R \approx -\tan^2\left(\frac{i}{2}\right) \cdot \{\sin(2\omega) \cos(2M) + \cos(2\omega) \sin(2M)\} \quad .$$

The value of R in degrees can be obtained by multiplying by $180^\circ/\pi$.

Finally, the ecliptic latitude b of the planets can also be handled in a similar manner. Taking equation (4.20)

$$z = r \cdot (\sin(\nu + \omega) \sin(i)) \quad (6.4)$$

for the z -coordinate as a function of the orbital elements and

$$\sin(b) = z/r \quad (6.5)$$

for small orbital inclinations i and ecliptic latitudes b , we have the approximation

$$b \approx i \sin(\nu + \omega) = \{i \sin \omega\} \cos(\nu) + \{i \cos \omega\} \sin(\nu) \quad .$$

If we replace ν by $\nu \approx M + 2e \sin M$, then the Taylor expansion gives

$$\cos \nu \approx \cos M - (2e \sin M) \sin M = \cos(M) - e(1 - \cos(2M))$$

$$\sin \nu \approx \sin M + (2e \sin M) \cos M = \sin(M) + e \sin(2M)$$

and thus

$$b \approx \{i \sin \omega\} \cdot \{\cos(M) - e(1 - \cos(2M))\} + \{i \cos \omega\} \cdot \{\sin(M) + e \sin(2M)\} \quad .$$

The most important components in a description of unperturbed planetary orbits using trigonometric series therefore are:

$$\begin{aligned} l = & \varpi + M \\ & + \frac{180^\circ}{\pi} \cdot \{2e\} \cdot \sin(M) \\ & + \frac{180^\circ}{\pi} \cdot \left\{ \frac{5}{4}e^2 - \tan^2\left(\frac{i}{2}\right) \cos(2\omega) \right\} \cdot \sin(2M) \end{aligned} \quad (6.6)$$

$$+ \frac{180^\circ}{\pi} \cdot \left\{ -\tan^2 \left(\frac{i}{2} \right) \sin(2\omega) \right\} \cdot \cos(2M)$$

$$\begin{aligned} b = & -\{ie \sin \omega\} \\ & + \{i \sin \omega\} \cos(M) + \{i \cos \omega\} \sin(M) \\ & + \{ie \sin \omega\} \cos(2M) + \{ie \cos \omega\} \sin(2M) \end{aligned} \quad (6.7)$$

$$r = \{a(1 + e^2/2)\} - \{ae\} \cdot \cos(M) - \{ae^2/2\} \cdot \cos(2M) \quad (6.8)$$

If specific values for the orbital elements are entered in these equations, then we obtain very compact formulae for ecliptic coordinates (l, b, r) as functions of the mean anomaly M . Taking Jupiter as an example, where

$$\begin{aligned} a &= 5.2 \text{ AU} & \Omega &= 100^\circ 0 & \omega &= 274^\circ 0 \\ e &= 0.048 & i &= 1^\circ 31 & \varpi &= 14^\circ 0 \end{aligned}$$

we obtain the following series:

$$\begin{aligned} l &= M + 14^\circ 0 + 19800'' \sin(M) + 620'' \sin(2M) + 4'' \cos(2M) \\ b &= +226'' - 4700'' \cos(M) + 329'' \sin(M) - 226'' \cos(2M) + 16'' \sin(2M) \\ r &= (5.206 - 0.250 \cos(M) - 0.006 \cos(2M)) \text{ AU} \end{aligned}$$

6.2 Perturbation Terms

The mutual perturbations of the planets, which are responsible for the departures from a keplerian orbit, result in additional terms in the series expansions, which contain both the anomaly of the perturbed planet and that of the perturbing planet. In the following, M_n indicates the mean anomaly of the n -th planet, M_5 being that of Jupiter, for example, and M_6 that of Saturn:

$$\begin{aligned} M_1 &= 0^\circ 4855407 + 415^\circ 2014314 \cdot T & (\text{Mercury}) \\ M_2 &= 0^\circ 1400197 + 162^\circ 5494552 \cdot T & (\text{Venus}) \\ M_3 &= 0^\circ 9931266 + 99^\circ 9973604 \cdot T & (\text{Sun, Earth}) \\ M_4 &= 0^\circ 0538553 + 53^\circ 1662736 \cdot T & (\text{Mars}) \\ M_5 &= 0^\circ 0565314 + 8^\circ 4302963 \cdot T & (\text{Jupiter}) \\ M_6 &= 0^\circ 8829867 + 3^\circ 3947688 \cdot T & (\text{Saturn}) \\ M_7 &= 0^\circ 3967117 + 1^\circ 1902849 \cdot T & (\text{Uranus}) \\ M_8 &= 0^\circ 7214906 + 0^\circ 6068526 \cdot T & (\text{Neptune}) \\ M_9 &= 0^\circ 0385795 + 0^\circ 4026667 \cdot T & (\text{Pluto}) \end{aligned} \quad (6.9)$$

The values are given here in multiples of an orbit (1°)¹. If they are multiplied by 360° or 2π , then we obtain the mean anomalies in the customary degrees

¹In some cases the values used in the programs differ slightly from the values given here. This is not an error, but a consequence of the different theories that form the basis of the series expansions.

Table 6.1. Periodic Terms in the Orbit of Jupiter

			$dl['']$		$dr[10^{-5} \text{ AU}]$		$db['']$		
i_5	i_T		cos	sin	cos	sin	cos	sin	
1	0		-113.1	19998.6	-25208.2	-142.2	-4670.7	288.9	Keplerian terms
1	1		-76.1	66.9	-84.2	-95.8	21.6	29.4	
1	2		-0.5	-0.3	0.4	-0.7	0.1	-0.1	
2	0		-3.4	632.0	-610.6	-6.5	-226.8	12.7	
2	1		-4.2	3.8	-4.1	-4.5	0.2	0.6	
3	0		-0.1	28.0	-22.1	-0.2	-12.5	0.7	
4	0		0.0	1.4	-1.0	0.0	-0.6	0.0	
i_5	i_6	i_T							
-1	-1	0	-0.2	1.4	2.0	0.6	0.1	-0.2	Saturn
0	-1	0	9.4	8.9	3.9	-8.3	-0.4	-1.4	
0	-2	0	5.6	-3.0	-5.4	-5.7	-2.0	0.0	
0	-3	0	-4.0	-0.1	0.0	5.5	0.0	0.0	
0	-5	0	3.3	-1.6	-1.6	-3.1	-0.5	-1.2	
1	-1	0	78.8	-14.5	11.5	64.4	-0.2	0.2	
1	-2	0	-2.0	-132.4	28.8	4.3	-1.7	0.4	
1	-2	1	-1.1	-0.7	0.2	-0.3	0.0	0.0	
1	-3	0	-7.5	-6.8	-0.4	-1.1	0.6	-0.9	
1	-4	0	0.7	0.7	0.6	-1.1	0.0	-0.2	
1	-5	0	51.5	-26.0	-32.5	-64.4	-4.9	-12.4	
1	-5	1	-1.2	-2.2	-2.7	1.5	-0.4	0.3	
2	-1	0	5.3	-0.7	0.7	6.1	0.2	1.1	
2	-2	0	-76.4	-185.1	260.2	-108.0	1.6	0.0	
2	-3	0	66.7	47.8	-51.4	69.8	0.9	0.3	
2	-3	1	0.6	-1.0	1.0	0.6	0.0	0.0	
2	-4	0	17.0	1.4	-1.8	9.6	0.0	-0.1	
2	-5	0	1066.2	-518.3	-1.3	-23.9	1.8	-0.3	
2	-5	1	-25.4	-40.3	-0.9	0.3	0.0	0.0	
2	-5	2	-0.7	0.5	0.0	0.0	0.0	0.0	
3	-2	0	-5.0	-11.5	11.7	-5.4	2.1	-1.0	
3	-3	0	16.9	-6.4	13.4	26.9	-0.5	0.8	
3	-4	0	7.2	-13.3	20.9	10.5	0.1	-0.1	
3	-5	0	68.5	134.3	-166.9	86.5	7.1	15.2	
3	-5	1	3.5	-2.7	3.4	4.3	0.5	-0.4	*
3	-6	0	0.6	1.0	-0.9	0.5	0.0	0.0	
3	-7	0	-1.1	1.7	-0.4	-0.2	0.0	0.0	
4	-2	0	-0.3	-0.7	0.4	-0.2	0.2	-0.1	
4	-3	0	1.1	-0.6	0.9	1.2	0.1	0.2	
4	-4	0	3.2	1.7	-4.1	5.8	0.2	0.1	
4	-5	0	6.7	8.7	-9.3	8.7	-1.1	1.6	
4	-6	0	1.5	-0.3	0.6	2.4	0.0	0.0	
4	-7	0	-1.9	2.3	-3.2	-2.7	0.0	-0.1	
4	-8	0	0.4	-1.8	1.9	0.5	0.0	0.0	
4	-9	0	-0.2	-0.5	0.3	-0.1	0.0	0.0	
4	-10	0	-8.6	-6.8	-0.4	0.1	0.0	0.0	
4	-10	1	-0.5	0.6	0.0	0.0	0.0	0.0	
5	-5	0	-0.1	1.5	-2.5	-0.8	-0.1	0.1	
5	-6	0	0.1	0.8	-1.6	0.1	0.0	0.0	
5	-9	0	-0.5	-0.1	0.1	-0.8	0.0	0.0	
5	-10	0	2.5	-2.2	2.8	3.1	0.1	-0.2	
i_5	i_7	i_T							
1	-1	0	0.4	0.9	0.0	0.0	0.0	0.0	Uranus
1	-2	0	0.4	0.4	-0.4	0.3	0.0	0.0	
i_5	i_6	i_7							
2	-6	3	-0.8	8.5	-0.1	0.0	0.0	0.0	Saturn and Uranus
3	-6	3	0.4	0.5	-0.7	0.5	-0.1	0.0	

or radians. Time is reckoned, as usual, in Julian centuries since epoch J2000:

$$T = (\text{JD} - 2451545)/36525 \quad .$$

The various perturbation terms can obviously be described in a table, such as Table 6.1, which applies to Jupiter. This shows the complete periodic variations in ecliptic longitude (dl) and latitude (db), as well as distance from the Sun (dr). Terms with equal arguments in the trigonometric series are brought together in a single line, which is of assistance in later calculations. Expressed in full, and taking the line marked with an asterisk (*) as an example, the values

$$\begin{aligned} dl &= 68''.5 \cdot \cos(3M_5 - 5M_6) + 134''.3 \cdot \sin(3M_5 - 5M_6) \\ dr &= (-166.9 \cdot \cos(3M_5 - 5M_6) + 86.5 \cdot \sin(3M_5 - 5M_6)) \cdot 10^{-5} \text{ AU} \\ db &= 7''.1 \cdot \cos(3M_5 - 5M_6) + 15''.2 \cdot \sin(3M_5 - 5M_6) \end{aligned}$$

are to be added to the longitude, latitude and radius. Terms that are additionally to be multiplied by T or T^2 , are indicated by 1 or 2 in the column headed i_T .

The influence of other planets is not confined to periodic perturbations but is also detectable in long-term (*secular*) variations in the mean orbital elements. This means that now components dependent on time T appear even in terms that actually describe purely Keplerian motion. In the JUP200 procedure the following equations are used to calculate Jupiter's coordinates:

$$\begin{aligned} l &= M_5 + 14^\circ 00' 07.6 + (5025''.2 + 0''.8T)T + dl \\ b &= 227''.3 - 0''.3T + db \\ r &= (5.208873 + 0.000041T)\text{AU} + dr \quad . \end{aligned}$$

where dl , db and dr are the complete periodic contributions to the orbit, and therefore include the Kepler terms and perturbations by Saturn and Uranus. The values of ecliptic longitude and latitude determined in this manner are referred to the *mean equinox of date*. The formula for l therefore contains the conspicuously large secular term $+5025''.2 \cdot T$, which reflects the motion of the vernal equinox.

Reasons of space forbid us from giving all the perturbation terms for every planet. Because the individual series expansions have a uniform structure, however, the details of Jupiter's orbit should suffice for readers to recognize the formulae and terms employed in the programs.

There are, nevertheless, differences in the way in which the Earth's orbit is handled from the treatment of the orbits of the other planets. The Earth orbits the Sun in company with the Moon, and one can imagine the two bodies being combined at their common centre of gravity (the barycentre). The motion of this centre of gravity (which moreover defines the mean ecliptic), can be envisaged as a perturbed Kepler orbit, like those of the other planets, and expressed as an appropriate series expansion. Because of the relative masses

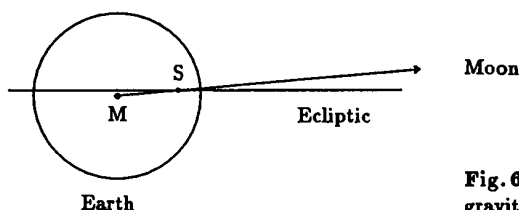


Fig. 6.1. The position of the centre of gravity of the Earth and the Moon

of the Earth and the Moon, the distance between the centre of the Earth and the barycentre is $1/81$ of the distance of the Moon from the Earth, and therefore amounts to about $3/4$ Earth radii (see Fig. 6.1). Seen from the Sun, the ecliptic longitude of the Earth therefore varies over the course of a month by as much as 7 arc-seconds from its mean position. At the same time the Earth's ecliptic latitude varies by 0.5 arc-second.

6.3 Numerical Treatment of the Series Expansions

In order to obtain accuracies that are of the order of seconds of arc, several hundred perturbation terms are required for each planet. If these were all specifically programmed, then not only would the code for the program be very long and unintelligible, but we would also have to accept unnecessarily long calculation times. This is because the evaluation of trigonometrical functions is relatively complex and slow when compared with simple computational processes such as addition and multiplication. Because of the large number of angular functions in the series expansions, this overhead becomes quite obvious even with fast computers.

We can, however, avoid a major part of this work, if we use the sine and cosine addition theorems. They enable angular functions to be calculated as the sum or difference of angles, the functions of the individual angles being known:

$$\begin{aligned}\cos(\alpha_1 + \alpha_2) &= \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 \\ \sin(\alpha_1 + \alpha_2) &= \sin \alpha_1 \cos \alpha_2 + \cos \alpha_1 \sin \alpha_2\end{aligned}\quad (6.10)$$

Using this method, we can very conveniently compute and store $\cos(iM)$ and $\sin(iM)$ for multiples $i = 2, 3, \dots$ of the mean anomaly M , from $\cos M$ and $\sin M$. This can be illustrated by a short piece of code:

```
VAR C,S: ARRAY[0..5] OF REAL;      (* Arrays for cos(i*M), sin(i*M) *)
    M : REAL;
    I : INTEGER;
PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
BEGIN
    C := C1*C2 - S1*S2;  (* cos(a1+a2)=cos(a1)cos(a2)-sin(a1)sin(a2) *)
    S := S1*C2 + C1*S2;  (* sin(a1+a2)=sin(a1)cos(a2)+cos(a1)sin(a2) *)
END;
```

```

BEGIN
  WRITE('M (rad) ?'); READLN(M); (* input mean anomaly M *)
  C[0] := 1.0; C[1] := COS(M); (* cos(0*M), cos(1*M) *)
  S[0] := 0.0; S[1] := SIN(M); (* sin(0*M), sin(1*M) *)
  FOR I:=2 TO 5 DO (* calculate cos(iM), sin(iM), i=2..5 *)
    ADDTHE ( C[1],S[1], C[I-1],S[I-1], C[I],S[I] );
  FOR I:=0 TO 5 DO (* output *)
    WRITELN( I:5, C[I]:12:7, S[I]:10:7 );
END;

```

The following example of the JUP200 procedure shows how all the perturbation terms can be evaluated and summed by this method. The **TERM** subroutine calculates the values of $u = \cos(i_5 M_5 + i_j M_j)$ and $v = \sin(i_5 M_5 + i_j M_j)$ with the indices i_5 and i_j , multiplies these with the corresponding coefficients and finally adds them to dl , dr and db . If we make the assumption that all terms with equal i_5 and i_j , but different $i_T = 0, 1, \dots$, may be evaluated one after the other, it suffices—as is done in **TERM**—to calculate u and v just for $i_T = 0$, and simply multiply by T each time to obtain $i_T = 1$ or $i_T = 2$. Although at first sight this may suggest a greater amount of labour, it soon pays off when—as in this case—one is working with a large number of terms.

```

(*-----*)
(* JUP200: Jupiter; ecliptic coordinates L,B,R (in deg and AU) *)
(* equinox of date *)
(* T: time in Julian centuries since J2000 *)
(* = (JED-2451545.0)/36525 *)
(*-----*)
PROCEDURE JUP200(T:REAL;VAR L,B,R:REAL);

CONST P2=6.283185307;
VAR C5,S5: ARRAY [-1..5] OF REAL;
    C,S: ARRAY [-10..0] OF REAL;
    M5,M6,M7: REAL;
    U,V,DL,DR,DB: REAL;
    I: INTEGER;

FUNCTION FRAC(X:REAL):REAL;
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
  BEGIN C:=C1+C2-S1*S2; S:=S1*C2+C1*S2; END;

PROCEDURE TERM(I5,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
  BEGIN
    IF IT=0 THEN ADDTHE(C5[I5],S5[I5],C[I],S[I],U,V)
    ELSE BEGIN U:=U*T; V:=V*T END;
    DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
  END;

PROCEDURE PERTSAT; (* Kepler terms and perturbations by Saturn *)
  VAR I: INTEGER;

```

BEGIN

```

C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M6); S[-1]:=-SIN(M6);
FOR I:=-1 DOWNT0 -9 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
TERM(-1, -1.0, -0.2, 1.4, 2.0, 0.6, 0.1, -0.2);
TERM( 0, -1.0, 9.4, 8.9, 3.9, -8.3, -0.4, -1.4);
TERM( 0, -2.0, 5.6, -3.0, -5.4, -5.7, -2.0, 0.0);
TERM( 0, -3.0, -4.0, -0.1, 0.0, 5.5, 0.0, 0.0);
TERM( 0, -5.0, 3.3, -1.6, -1.6, -3.1, -0.5, -1.2);
TERM( 1, 0.0, -113.1, 19998.6, -25208.2, -142.2, -4670.7, 288.9);
TERM( 1, 0.1, -76.1, 66.9, -84.2, -95.8, 21.6, 29.4);
TERM( 1, 0.2, -0.5, -0.3, 0.4, -0.7, 0.1, -0.1);
TERM( 1, -1.0, 78.8, -14.5, 11.5, 64.4, -0.2, 0.2);
TERM( 1, -2.0, -2.0, -132.4, 28.8, 4.3, -1.7, 0.4);
TERM( 1, -2.1, -1.1, -0.7, 0.2, -0.3, 0.0, 0.0);
TERM( 1, -3.0, -7.5, -6.8, -0.4, -1.1, 0.6, -0.9);
TERM( 1, -4.0, 0.7, 0.7, 0.6, -1.1, 0.0, -0.2);
TERM( 1, -5.0, 51.5, -26.0, -32.5, -64.4, -4.9, -12.4);
TERM( 1, -5.1, -1.2, -2.2, -2.7, 1.5, -0.4, 0.3);
TERM( 2, 0.0, -3.4, 632.0, -610.6, -6.5, -226.8, 12.7);
TERM( 2, 0.1, -4.2, 3.8, -4.1, -4.5, 0.2, 0.6);
TERM( 2, -1.0, 5.3, -0.7, 0.7, 6.1, 0.2, 1.1);
TERM( 2, -2.0, -76.4, -185.1, 260.2, -108.0, 1.6, 0.0);
TERM( 2, -3.0, 66.7, 47.8, -51.4, 69.8, 0.9, 0.3);
TERM( 2, -3.1, 0.6, -1.0, 1.0, 0.6, 0.0, 0.0);
TERM( 2, -4.0, 17.0, 1.4, -1.8, 9.6, 0.0, -0.1);
TERM( 2, -5.0, 1066.2, -518.3, -1.3, -23.9, 1.8, -0.3);
TERM( 2, -5.1, -25.4, -40.3, -0.9, 0.3, 0.0, 0.0);
TERM( 2, -5.2, -0.7, 0.5, 0.0, 0.0, 0.0, 0.0);
TERM( 3, 0.0, -0.1, 28.0, -22.1, -0.2, -12.5, 0.7);
TERM( 3, -2.0, -5.0, -11.5, 11.7, -5.4, 2.1, -1.0);
TERM( 3, -3.0, 16.9, -6.4, 13.4, 26.9, -0.5, 0.8);
TERM( 3, -4.0, 7.2, -13.3, 20.9, 10.5, 0.1, -0.1);
TERM( 3, -5.0, 68.5, 134.3, -166.9, 86.5, 7.1, 15.2);
TERM( 3, -5.1, 3.5, -2.7, 3.4, 4.3, 0.5, -0.4);
TERM( 3, -6.0, 0.6, 1.0, -0.9, 0.5, 0.0, 0.0);
TERM( 3, -7.0, -1.1, 1.7, -0.4, -0.2, 0.0, 0.0);
TERM( 4, 0.0, 0.0, 1.4, -1.0, 0.0, -0.6, 0.0);
TERM( 4, -2.0, -0.3, -0.7, 0.4, -0.2, 0.2, -0.1);
TERM( 4, -3.0, 1.1, -0.6, 0.9, 1.2, 0.1, 0.2);
TERM( 4, -4.0, 3.2, 1.7, -4.1, 5.8, 0.2, 0.1);
TERM( 4, -5.0, 6.7, 8.7, -9.3, 8.7, -1.1, 1.6);
TERM( 4, -6.0, 1.5, -0.3, 0.6, 2.4, 0.0, 0.0);
TERM( 4, -7.0, -1.9, 2.3, -3.2, -2.7, 0.0, -0.1);
TERM( 4, -8.0, 0.4, -1.8, 1.9, 0.5, 0.0, 0.0);
TERM( 4, -9.0, -0.2, -0.5, 0.3, -0.1, 0.0, 0.0);
TERM( 4, -10.0, -8.6, -6.8, -0.4, 0.1, 0.0, 0.0);
TERM( 4, -10.1, -0.5, 0.6, 0.0, 0.0, 0.0, 0.0);
TERM( 5, -5.0, -0.1, 1.5, -2.5, -0.8, -0.1, 0.1);
TERM( 5, -6.0, 0.1, 0.8, -1.6, 0.1, 0.0, 0.0);
TERM( 5, -9.0, -0.5, -0.1, 0.1, -0.8, 0.0, 0.0);
TERM( 5, -10.0, 2.5, -2.2, 2.8, 3.1, 0.1, -0.2);

```

END;

```

PROCEDURE PERTURA; (* perturbations by Uranus *)
BEGIN
  C[-1]:=COS(M7); S[-1]:=-SIN(M7);
  ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
  TERM( 1, -1,0, 0.4, 0.9, 0.0, 0.0, 0.0, 0.0);
  TERM( 1, -2,0, 0.4, 0.4, -0.4, 0.3, 0.0, 0.0);
END;

PROCEDURE PERTSUR; (* perturbations by Saturn and Uranus *)
VAR PHI,X,Y: REAL;
BEGIN
  PHI:=(2*M5-6*M6+3*M7);
  X:=COS(PHI); Y:=SIN(PHI);
  DL:=DL-0.8*X+8.5*Y; DR:=DR-0.1*X;
  ADDTHE(X,Y,C5[1],S5[1],X,Y);
  DL:=DL+0.4*X+0.5*Y; DR:=DR-0.7*X+0.5*Y; DB:=DB-0.1*X;
END;

BEGIN (* JUP200 *)

  DL:=0.0; DR:=0.0; DB:=0.0;
  M5:=P2*FRAC(0.0565314+8.4302963*T); M6:=P2*FRAC(0.8829867+3.3947688*T);
  M7:=P2*FRAC(0.3969537+1.1902586*T);
  C5[0]:=1.0; S5[0]:=0.0;
  C5[1]:=COS(M5); S5[1]:=SIN(M5); C5[-1]:=C5[1]; S5[-1]:=-S5[1];
  FOR I:=2 TO 5 DO ADDTHE(C5[I-1],S5[I-1],C5[1],S5[1],C5[I],S5[I]);
  PERTSAT; PERTURA; PERTSUR;
  L:= 360.0*FRAC(0.0388910 + M5/P2 + ((5025.2+0.8*T)*T+DL)/1296.0E3 );
  R:= 5.208873 + 0.000041*T + DR*1.0E-5;
  B:= ( 227.3 - 0.3*T + DB ) / 3600.0;

END; (* JUP200 *)

(*-----*)

```

For a given time T , JUP200 and the MER200, VEN200, ..., PLU200 procedures given in the Appendix return the heliocentric longitude l , latitude b and radius r of the planet concerned, relative to the ecliptic and vernal equinox of date. The time is measured in Julian centuries since epoch J2000:

$$T = (\text{JD} - 2451545)/36525 \quad .$$

In calculating the Julian Date JD, care must be taken to enter Ephemeris Time ET (or Dynamical Time TDT/TDB) rather than Universal Time UT (see Chap. 3.4). An appropriate routine for calculating geocentric solar coordinates has already been given in Chap. 2.

For the 19th and 20th centuries the accuracy of the programs is within a few seconds of arc (see Table 6.2). For more remote dates, one must expect less accurate results. First, the basic theories were developed around the turn of the century and therefore could only be devised to fit the observations

Table 6.2. Mean errors of the programs MER200...PLU200 for the period 1750–2250 or 1890–2100 (Pluto)

Planet	$\Delta l ["]$	$\Delta b ["]$	$\Delta r [AU]$
Mercury	1.0 - 1.5	0.7 - 1.0	1.0 - $1.5 \cdot 10^{-6}$
Venus	0.5 - 1.0	0.2 - 2.5	0.5 - $1.5 \cdot 10^{-6}$
Sun Earth	0.5 - 2.0	0.0 - 0.1	0.4 - $1.5 \cdot 10^{-6}$
Mars	0.5 - 2.5	0.1 - 1.0	3 - $10 \cdot 10^{-6}$
Jupiter	2 - 8	0.7 - 1.0	20 - $30 \cdot 10^{-6}$
Saturn	2 - 11	0.8 - 1.5	40 - $100 \cdot 10^{-6}$
Uranus	3 - 8	0.7 - 1.0	50 - $200 \cdot 10^{-6}$
Neptune	3 - 40	1.0 - 2.0	500 - $1000 \cdot 10^{-6}$
Pluto	1 - 9	0.2 - 2.2	200 - $1000 \cdot 10^{-6}$

available at the time. Second, perturbation terms that increase as T^3 have been neglected in the programs. Currently these values are very small ($< 0''.1$), but over historical periods of time they can easily increase to one thousand times this value.

6.4 Apparent and Astrometric Coordinates

If one consults a planetary ephemeris in a good yearbook, one usually finds details of whether the values included are *geometric*, *apparent*, or *astrometric* coordinates. The difference between these three variants is negligible, as long as one is only concerned with accuracies of 1 arc-minute or worse. In order to really use the accuracy of the programs given in the last section, however, we need to recognize and understand the most important corrections to the planetary coordinates.

While geometric coordinates indicate the point in space at which a planet is *located* at a specific time t , the apparent and astrometric coordinates indicate the direction in which the planet is *observed*.

Geometric coordinates are normally chosen only to express heliocentric planetary positions. The coordinate system being used is unambiguously determined by giving the equinox. The set of procedures MER200...PLU200 yields *geometric heliocentric ecliptic coordinates referred to the equinox of date* (i.e., of the calculated epoch). Naturally this does not apply to SUN200, which calculates the corresponding *geocentric* position of the Sun.

In contrast, geocentric positions (except distances) are generally given in the form of astrometric or apparent positions. The reason for this is that because of its motion and light-time, the planet is not observed at the position it occupies at the calculated time. This effect has already been mentioned in connection with calculating cometary orbits. Because the length of

the path actually travelled by the light is not directly observable, geocentric ephemerides are also normally restricted to giving the geometric distance.

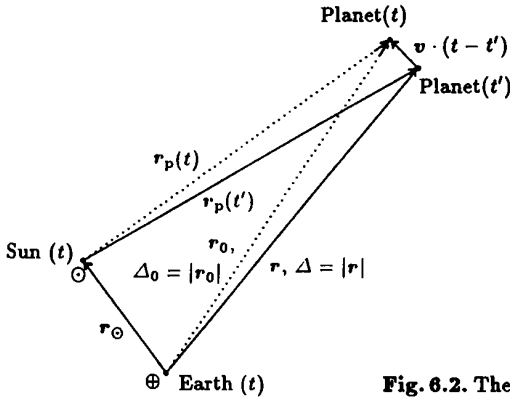


Fig. 6.2. The motion of a planet during light-time

6.4.1 Aberration and Light-Time

Astrometric coordinates indicate the direction from which an observed ray of light from a planet has reached the Earth. This ray of light links the position of the Earth at the time of observation t and the position of the planet at the time t' at which the light was emitted. The light-time $\tau = t - t'$ is given approximately by the geometric distance

$$\Delta_0 = |\mathbf{r}_0| \quad \text{where} \quad \mathbf{r}_0 = \mathbf{r}_\odot(t) + \mathbf{r}_p(t) \quad (6.11)$$

whence $\tau \approx \Delta_0/c$ (see Fig. 6.2 for the definition of the individual quantities). Because the planet moves in approximately a straight line during this time, the astrometric coordinates of a planet at the time of observation t are given by the expression

$$\begin{aligned} \mathbf{r} &= \mathbf{r}_\odot(t) + \mathbf{r}_p(t') \\ &\approx \mathbf{r}_\odot(t) + \left\{ \mathbf{r}_p(t) - \mathbf{v}_p(t) \cdot \frac{\Delta_0}{c} \right\} \end{aligned} \quad (6.12)$$

The difference between the *astrometric* coordinates $\mathbf{r}$ and geometric coordinates $\mathbf{r}_0$ is known as *light-time correction*. Astrometric coordinates can be directly marked on a celestial atlas or can be compared with known star positions on a photographic plate.

If, however, we point an equatorially mounted telescope at a planet, using astrometric coordinates for setting the circles, then we find a difference between the setting position and the observed position. This difference amounts to about 20 arc-seconds. This error is caused because astrometric coordinates indicate the direction of a ray of light falling on the Earth in a reference system that is *at rest* with respect to the Sun. However, the observer is subject to

the daily rotation of the Earth and the yearly revolution of the Earth around the Sun. Because of these motions and the finite velocity of light the apparent direction in which the light arrives from the planet is altered. A frequently quoted example of the way in which appearances differ between a moving observer and one at rest can easily be observed by anybody: according to the velocity of an observer in a vehicle, raindrops that are falling vertically towards the ground appear to him to be coming more or less from the direction in which he is moving.

The coordinate correction required in changing from a stationary to a moving reference system also affects the fixed stars as well as the planets and comets. We therefore also speak of *stellar aberration*. Star positions in catalogues and celestial atlases are given as they would appear to an observer stationary in the Solar System. They therefore correspond to the astrometric coordinates of the planets, with which they may be directly compared. An observer on Earth, on the other hand, using divided circles, measures apparent stellar positions, which vary periodically over the course of the year from the catalogued positions.

A rigorous description of stellar aberration would take us rather too far, because it actually is a phenomenon governed by special relativity. As the velocity of the Earth is relatively small when compared with the velocity of light, however, the semi-classical description that follows produces the same result within the degree of accuracy that concerns us here.

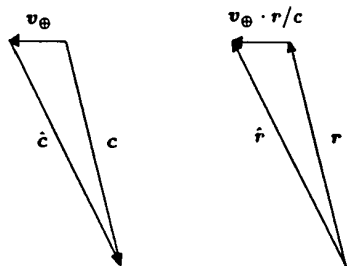


Fig. 6.3. The effect of stellar aberration

If $\mathbf{r}$ is the astrometric-coordinate vector of the planet, then the ray of light falling on the Earth may be described by the light-velocity vector

$$\mathbf{c} = -c \cdot \frac{\mathbf{r}}{|\mathbf{r}|}$$

(see Fig. 6.3). We obtain the corresponding vector $\hat{\mathbf{c}}$ in the observer's Earth-centred system if we subtract the velocity $\mathbf{v}_\oplus$ of the Earth relative to the Sun:

$$\hat{\mathbf{c}} = \mathbf{c} - \mathbf{v}_\oplus$$

An observer therefore sees the planet in the direction

$$\hat{\mathbf{r}} = -|\mathbf{r}| \cdot \frac{\hat{\mathbf{c}}}{c} = \mathbf{r} + \mathbf{v}_\oplus \cdot \frac{|\mathbf{r}|}{c} \quad (6.13)$$

$$\approx \mathbf{r} + \mathbf{v}_{\oplus} \cdot \frac{\Delta_0}{c} .$$

The coordinates thus determined are known as the *apparent* position of the planet, if the coordinate system chosen is additionally oriented in exactly the same direction as the current position of the Earth's axis. The choice of equinox required for this to be the case, and the allowance for nutation are the subject of the next section but one.

6.4.2 Velocities of the Planets

We must next consider how to calculate the heliocentric velocity vector for an individual planet, which is required to take account of the various aberration effects.

If we differentiate ecliptic coordinates (l, b, r) with respect to time and only retain the major terms, then the derivatives $(\dot{l}, \dot{b}, \dot{r})$ are obtained as short trigonometric series, which depend only on the mean anomaly M of the planet concerned:

$$\begin{aligned} \dot{l} &= \dot{M} \cdot (1 + 2e \cos(M) + 5/4 e^2 \cos(2M) + \dots) \\ \dot{b} &= i \dot{M} \cdot (-\sin(\omega) \sin(M) + \cos(\omega) \cos(M) + \dots) \\ \dot{r} &= a \dot{M} \cdot (e \sin(M) + e^2/2 \sin(2M) + \dots) \end{aligned} \quad (6.14)$$

The total number of terms included follows from the orbital eccentricity and inclination. For Jupiter, for example, only the following terms are required:

$$\begin{aligned} \dot{l} &= (+14.50 + 1.41 \cos(M)) \cdot 10^{-4} \text{ rad/d} \\ \dot{b} &= 0.33 \sin(M) \cdot 10^{-4} \text{ rad/d} \\ \dot{r} &= 3.66 \sin(M) \cdot 10^{-4} \text{ AU/d} \end{aligned}$$

From the time derivatives of the polar coordinates $(\dot{l}, \dot{b}, \dot{r})$ we can now derive the components of the velocity vector in ecliptic coordinates:

$$\begin{aligned} \dot{x} &= \dot{r} \cdot \cos l \cos b - \dot{l} \cdot r \sin l \cos b - \dot{b} \cdot r \cos l \sin b \\ \dot{y} &= \dot{r} \cdot \sin l \cos b + \dot{l} \cdot r \cos l \cos b - \dot{b} \cdot r \sin l \sin b \\ \dot{z} &= \dot{r} \cdot \sin b + \dot{b} \cdot r \cos b \end{aligned} \quad (6.15)$$

All the steps required to calculate geocentric coordinates taking light-time and aberration into account are combined in the **GEOCEN** procedure. Depending on the value of the variable **IMODE**, corrections for planetary (**IMODE**=1,2) and stellar aberration (**IMODE**=2) are applied to the geometric coordinates. In addition, the geometric, geocentric distance Δ_0 is always given.

```

(*-----*)
(*)
(*) GEOCEN: geocentric coordinates (geometric and light-time corrected) (*)
(*)
(*) T:          time in Julian centuries since J2000 (*)
(*)           T=(JD-2451545.0)/36525.0 (*)
(*) LP,BP,RP: ecliptic heliocentric coordinates of the planet (*)
(*) LS,BS,RS: ecliptic geocentric coordinates of the sun (*)
(*)
(*) IPLAN:    planet (0=Sun,1=Mercury,2=Venus,3=Earth,...,9=Pluto) (*)
(*) IMODE:    desired type of coordinates (see description of X,Y,Z) (*)
(*)           (0=geometric,1=astrometric,2=apparent) (*)
(*) XP,YP,ZP: ecliptic heliocentric coordinates of the planet (*)
(*) XS,YS,ZS: ecliptic geocentric coordinates of the Sun (*)
(*) X, Y, Z : ecliptic geocentric coordinates of the planet (geometric) (*)
(*)             if IMODE=0, astrometric if IMODE=1, apparent if IMODE=2) (*)
(*) DELTA0:   geocentric distance (geometric) (*)
(*)
(*) (all angles in degrees, distances in AU) (*)
(*)
(*-----*)

```

```

PROCEDURE GEOCEN(T, LP,BP,RP, LS,BS,RS: REAL; IPLAN,IMODE: INTEGER;
  VAR XP,YP,ZP, XS,YS,ZS, X,Y,Z,DELTA0: REAL);

```

```

  CONST P2=6.283185307;

```

```

  VAR DL,DB,DR, DLS,DBS,DRS, FAC: REAL;
  VI,VY,VZ, VXS,VYS,VZS, M : REAL;

```

```

  FUNCTION FRAC(X:REAL):REAL;
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;

```

```

  PROCEDURE POSVEL(L,B,R,DL,DB,DR: REAL; VAR X,Y,Z,VI,VY,VZ:REAL);
  VAR CL,SL,CB,SB: REAL;
  BEGIN
    CL:=CS(L); SL:=SN(L); CB:=CS(B); SB:=SN(B);
    X := R*CL*CB; VI := DR*CL*CB-DL*R*SL*CB-DB*R*CL*SB;
    Y := R*SL*CB; VY := DR*SL*CB+DL*R*CL*CB-DB*R*SL*SB;
    Z := R*SB; VZ := DR*SB +DB*R*CB;
  END;

```

```

  BEGIN

```

```

    DL:=0.0; DB:=0.0; DR:=0.0; DLS:=0.0; DBS:=0.0; DRS:=0.0;

```

```

    IF (IMODE>0) THEN

```

```

      BEGIN

```

```

        M := P2*FRAC(0.9931266+ 99.9973604*T); (* Sun *)
        DLS := 172.00+5.75*SIN(M); DRS := 2.87*COS(M); DBS := 0.0;

```

(* dl,db in 1e-4 rad/d, dr in 1e-4 AU/d *)

CASE IPLAN OF

0: BEGIN DL:=0.0; DB:=0.0; DR:=0.0; END; (* Sun *)

1: BEGIN (* Mercury *)

M := P2*FRAC(0.4855407+415.2014314*T);

DL := 714.00+292.88*COS(M)+71.98*COS(2*M)+18.16*COS(3*M)
+4.61*COS(4*M)+3.81*SIN(2*M)+2.43*SIN(3*M)
+1.08*SIN(4*M);

DR := 55.94*SIN(M)+11.36*SIN(2*M)+2.80*SIN(3*M);

DB := 73.40*COS(M)+29.82*COS(2*M)+10.22*COS(3*M)
+3.28*COS(4*M)-40.44*SIN(M)-18.55*SIN(2*M)
-5.58*SIN(3*M)-1.72*SIN(4*M);

END;

2: BEGIN (* Venus *)

M := P2*FRAC(0.1400197+162.5494552*T);

DL := 280.00+3.79*COS(M); DR := 1.37*SIN(M);

DB := 9.64*COS(M)-13.57*SIN(M);

END;

3: BEGIN DL:=DLS; DR:=DRS; DB:=-DBS; END; (* Earth *)

4: BEGIN (* Mars *)

M := P2*FRAC(0.0638553+53.1662736*T);

DL := 91.50+17.07*COS(M)+2.03*COS(2*M);

DR := 12.98*SIN(M)+1.21*COS(2*M);

DB := 0.83*COS(M)+2.80*SIN(M);

END;

5: BEGIN (* Jupiter *)

M := P2*FRAC(0.0565314+8.4302983*T);

DL := 14.50+1.41*COS(M); DR:=3.66*SIN(M); DB:=0.33*SIN(M);

END;

6: BEGIN (* Saturn *)

M := P2*FRAC(0.8829867+3.3947888*T);

DL := 5.84+0.65*COS(M); DR:=3.09*SIN(M); DB:=0.24*COS(M);

END;

7: BEGIN (* Uranus *)

M := P2*FRAC(0.3967117+1.1902849*T);

DL := 2.05+0.19*COS(M); DR:=1.88*SIN(M); DB:=-0.03*SIN(M);

END;

8: BEGIN (* Neptune *)

M := P2*FRAC(0.7214906+0.8068526*T);

DL := 1.04+0.02*COS(M); DR:=0.27*SIN(M); DB:=0.03*SIN(M);

END;

9: BEGIN (* Pluto *)

M := P2*FRAC(0.0385795+0.4026667*T);

DL := 0.69+0.34*COS(M)+0.12*COS(2*M)+0.05*COS(3*M);

DR := 6.66*SIN(M)+1.84*SIN(2*M);

DB := -0.08*COS(M)-0.17*SIN(M)-0.09*SIN(2*M);

END;

END;

END;

POSVEL (LS,BS,RS,DLS,DBS,DRS,IS,YS,ZS,VIS,VYS,VZS);

POSVEL (LP,BP,RP,DL,DB,DR,XP,YP,ZP,VX,VY,VZ);

X:=XP+XS; Y:=YP+YS; Z:=ZP+ZS; DELTA0 := SQRT(X*X+Y*Y+Z*Z);

```
IF IPLAN=3 THEN BEGIN X:=0.0; Y:=0.0; Z:=0.0; DELTA0:=0.0 END;
```

```
FAC := 0.00578 * DELTA0 * 1E-4;
```

```
CASE IMODE OF
```

```
1: BEGIN X:=X-FAC*VX; Y:=Y-FAC*VY; Z:=Z-FAC*VZ; END;
```

```
2: BEGIN X:=X-FAC*(VX+VXS); Y:=Y-FAC*(VY+VYS); Z:=Z-FAC*(VZ+VZS);END;
```

```
END;
```

```
END;
```

```
(*-----*)
```

6.4.3 Nutation

Apparent coordinates are coordinates that are designed to be used with a telescope fitted with graduated circles. Because the telescope's mounting is always rigidly pointing in the same direction as the Earth's axis, the coordinate system employed must have the corresponding orientation. This chiefly demands that the apparent coordinates always relate to the equinox of date, i.e., to the current position of the equator, ecliptic, and vernal equinox. Apart from precession, there is another effect on the position of the Earth's axis to be taken into account, that has not yet been mentioned. While precession indicates the secular—i.e., the long-term—position of the Earth's axis, this additional, small, periodic variation is *nutation*. As a result of nutation, the Earth's *true* pole rotates once in every 18.6 years around the *mean* pole, the motion of which is described by precession. The nutation period is determined by the time required for the ascending node Ω of the Moon's orbit to complete one rotation. Nutation in longitude ($\Delta\psi$), i.e., the difference in longitude between the true and mean vernal equinoxes, has an amplitude of about 17 arc-seconds. The obliquity of the ecliptic simultaneously varies by $\Delta\epsilon \approx 9''$ (see Fig. 6.4).

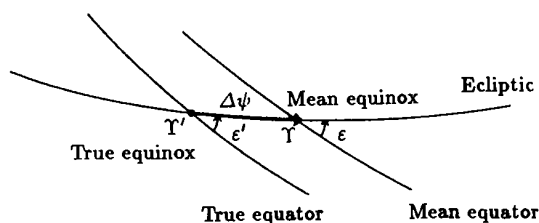


Fig. 6.4. The shift in the positions of the equator, the ecliptic and the vernal equinox, caused by nutation

Both of these quantities consist of around one hundred individual terms. For most purposes, however, only a few are required:

$$\begin{aligned}\Delta\psi &= -17''.200 \cdot \sin(\Omega) + 0''.206 \cdot \sin(2\Omega) + 0''.143 \cdot \sin(I') \\ &\quad - 1''.319 \cdot \sin(2(F - D + \Omega)) - 0''.227 \cdot \sin(2(F + \Omega)) \quad (6.16) \\ \Delta\epsilon &= +9''.203 \cdot \cos(\Omega) - 0''.090 \cdot \cos(2\Omega) \\ &\quad + 0''.574 \cdot \cos(2(F - D + \Omega)) + 0''.098 \cdot \cos(2(F + \Omega))\end{aligned}$$

Apart from the longitude of the ascending node Ω , these terms depend on various additional arguments (F, D, l'), derived from the mean longitudes and anomalies of the Sun and Moon (see (8.2)...(8.5)). The true equatorial coordinates (x', y', z') are obtained from the mean coordinates (x, y, z), used hitherto, by adding a small correction ($\Delta x, \Delta y, \Delta z$):

$$\begin{aligned}\Delta x &= -(y \cos \varepsilon + z \sin \varepsilon) \cdot \Delta \psi \\ \Delta y &= x \cos \varepsilon \cdot \Delta \psi - z \cdot \Delta \varepsilon \\ \Delta z &= x \sin \varepsilon \cdot \Delta \psi + y \cdot \Delta \varepsilon\end{aligned}\quad (6.17)$$

Here ε is the mean obliquity of the ecliptic from (2.5).

The NUTEQU procedure takes as input the mean equatorial coordinates of a planet and turns these into true coordinates with respect to the time T . In the process the input data are overwritten.

```
(*-----*)
(* NUTEQU: transformation of mean to true coordinates *)
(* (including terms >0.1" according to IAU 1980) *)
(* T = (JD-2451545.0)/36525.0 *)
(*-----*)
PROCEDURE NUTEQU(T:REAL;VAR X,Y,Z:REAL);

  CONST ARC=208284.8062;      (* arcseconds per radian = 3800*180/pi *)
        P2 =6.283185307;      (* 2*pi *)

  VAR  LS,D,F,N,EPS : REAL;
        DPSI,DEPS,C,S: REAL;
        DX,DY,DZ : REAL;
  FUNCTION FRAC(X:REAL):REAL;
    (* with several compilers it may be necessary to replace TRUNC *)
    (* by LONG_TRUNC or INT if T<-24! *)
    BEGIN FRAC:=X-TRUNC(X) END;
  BEGIN
    LS := P2*FRAC(0.993133+ 99.997306*T); (* mean anomaly Sun *)
    D := P2*FRAC(0.827382+1236.853087*T); (* diff. longitude Moon-Sun *)
    F := P2*FRAC(0.259089+1342.227826*T); (* mean argument of latitude *)
    N := P2*FRAC(0.347346- 5.372447*T); (* longit. ascending node *)
    EPS := 0.4090928-2.2696E-4*T;      (* obliquity of the ecliptic *)
    DPSI := ( -17.200*SIN(N) - 1.319*SIN(2*(F-D+N)) - 0.227*SIN(2*(F+N))
              + 0.206*SIN(2*N) + 0.143*SIN(LS) ) / ARC;
    DEPS := ( + 9.203*COS(N) + 0.574*COS(2*(F-D+N)) + 0.098*COS(2*(F+N))
              - 0.090*COS(2*N) ) / ARC;
    C := DPSI*COS(EPS); S := DPSI*SIN(EPS);
    DX := -(C*Y+S*Z); DY := (C*X-DEPS*Z); DZ := (S*X+DEPS*Y);
    X := X + DX; Y := Y + DY; Z := Z + DZ;
  END;
(*-----*)
```

6.5 The PLANPOS Program

The PLANPOS program calculates the position of the Sun and of the nine major planets for a given date. Because of the restricted validity of the PLU200 procedure, Pluto's coordinates are given only between the years 1890 and 2100. As well as the date, the equinox to which the calculated positions are referred can be specified. There are three choices: apparent coordinates and astrometric coordinates for either of the two equinoxes currently employed, B1950 and J2000. Apparent coordinates are referred to the true equinox of date and contain corrections for precession, nutation, aberration and light-time. They can, for example, be either set on the circles of an equatorially mounted telescope, or read from such circles. Astrometric coordinates are required if we want to plot planetary positions on a star chart for the appropriate equinox. They take precession and light-time into account.

```
(*-----*)
(*                               PLANPOS                               *)
(*      heliocentric and geocentric planetary positions                *)
(*                               version 93/07/01                        *)
(*-----*)
PROGRAM PLANPOS(INPUT,OUTPUT);

  CONST J2000 = 0.0;
        B1950 = -0.500002108;
  TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
  VAR  DAY, MONTH, YEAR, IPLAN, IMODE, K      : INTEGER;
        HOUR, MODJD, T, TEQX                  : REAL;
        X,Y,Z, XP,YP,ZP, XS,YS,ZS             : REAL;
        L,B,R, LS,BS,RS, RA,DEC,DELTA,DELTAO : REAL;
        A                                      : REAL33;
        MODE                                   : CHAR;

(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS, TN, ATN, ATN2, POLAR, DMS, MJD                                         *)
(* ECLEQU, PMATECL, PRECART, NUTEQU                                              *)
(* MER200, VEN200, MAR200, JUP200, SAT200, URA200, NEP200, PLU200                *)
(* SUN200, GEOCEN                                                                *)
(*-----*)

(*-----*)
(* PRINTOUT: print coordinates of one planet                                     *)
(*-----*)
PROCEDURE PRINTOUT(IPLAN:INTEGER;L,B,R,RA,DEC,DELTA:REAL);
  VAR H,M: INTEGER;
      S : REAL;
  BEGIN
    DMS(L,H,M,S);  WRITE (H:3,M:3,S:5:1);
    DMS(B,H,M,S);  WRITE (H:4,M:3,S:5:1);
    IF IPLAN<4 THEN WRITE (R:11:6)
                  ELSE WRITE (R:10:5,' ');
```



```
DMS(RA,H,M,S); WRITE (H:4,M:3,S:6:2);
DMS(DEC,H,M,S); WRITE (H:4,M:3,S:5:1);
IF IPLAN<4 THEN WRITE (DELTA:11:6)
      ELSE WRITE (DELTA:10:5,' ');
```

```
END;
```

```
(*-----*)
```

```
BEGIN (* PLANPOS *)
```

```
WRITELN;
WRITELN(' PLANPOS: geocentric and heliocentric planetary positions ');
WRITELN(' version 93/07/01 ');
WRITELN(' (c) 1993 Thomas Pfleger, Oliver Montenbruck ');
WRITELN;
```

```
REPEAT
```

```
WRITELN;
WRITELN(' (J) J2000 astrometric (B) B1950 astrometric ');
WRITELN(' (A) apparent coordinates (E) end ');
WRITELN;
WRITE (' enter option: '); READLN (MODE);
WRITELN;
```

```
IF MODE IN ['A','a','J','j','B','b'] THEN BEGIN
```

```
(* read date *)
WRITE (' date (year month day hour) ? ');
READLN (YEAR,MONTH,DAY,HOUR); WRITELN; WRITELN; WRITELN;
MODJD := MJD (DAY,MONTH,YEAR,HOUR); T:=(MODJD-51544.5)/38525.0;
WRITE (' date: ', YEAR:4,'/',MONTH:2,'/',DAY:2,' ',
      HOUR:5:1,'(ET)');
WRITE ('JD:':6,(MODJD+2400000.5):12:3,'equinox ':18);
CASE MODE OF
  'A','a': WRITELN ('of date');
  'J','j': WRITELN ('J2000');
  'B','b': WRITELN ('B1950');
END;
WRITELN;
```

```
(* header *)
WRITE (' ':10,'l':6,'b':12,'r':11);
WRITELN (' ':7,'RA':5,'Dec':13,'delta':13);
WRITE (' ':9,' o '' '' ',':3,' o '' '' ',':6,'AU',':4);
WRITELN (' ':2,' h m s',':4,' o '' '' ',':6,'AU');
```

```
(* ecliptic coordinates of the sun, Equinox T *)
SUN200 (T,LS,BS,RS);
```

```
(* planetary coordinates; include Pluto between 1890 and 2100 *)
IF ( (-1.1<T) AND (T<+1.0) ) THEN K:=9 ELSE K:=8;
FOR IPLAN:=0 TO K DO
```

```

BEGIN
  (* heliocentric ecliptic coordinates of the planet *)
  CASE IPLAN OF
    1: MER200(T,L,B,R); 2: VEN200(T,L,B,R);
    4: MAR200(T,L,B,R); 5: JUP200(T,L,B,R); 6: SAT200(T,L,B,R);
    7: URA200(T,L,B,R); 8: NEP200(T,L,B,R); 9: PLU200(T,L,B,R);
    0: BEGIN L:=0.0; B:=0.0; R:=0.0; END;
    3: BEGIN L:=LS+180.0; B:=-BS; R:=RS; END;
  END;
  (* geocentric ecliptic coordinates (light-time corrected) *)
  IF MODE IN ['A','a'] THEN IMODE:=2 ELSE IMODE:=1;
  GEOCEN (T, L,B,R, LS,BS,RS, IPLAN,IMODE,
    XP,YP,ZP, XS,YS,ZS, X,Y,Z,DELTA0);
  (* precession, equatorial coordinates, nutation *)
  CASE MODE OF
    'J','j': TEQX:=J2000; 'B','b': TEQX:=B1950;
  END;
  IF MODE IN ['A','a']
  THEN
    BEGIN
      ECLEQU(T,X,Y,Z); NUTEQU(T,X,Y,Z);
    END
  ELSE
    BEGIN
      PMATECL(T,TEQX,A); PRECART (A,XP,YP,ZP);
      PRECART (A,X,Y,Z); ECLEQU (TEQX,X,Y,Z);
    END;
  (* spherical coordinates *)
  POLAR (XP,YP,ZP,R,B,L);
  POLAR (X,Y,Z,DELTA,DEC,RA); RA:=RA/15.0;
  (* output *)
  CASE IPLAN OF
    0: WRITE(' Sun '); 1: WRITE(' Mercury ');
    2: WRITE(' Venus '); 3: WRITE(' Earth ');
    4: WRITE(' Mars '); 5: WRITE(' Jupiter ');
    6: WRITE(' Saturn '); 7: WRITE(' Uranus ');
    8: WRITE(' Neptune '); 9: WRITE(' Pluto ');
  END;
  PRINTOUT(IPLAN,L,B,R,RA,DEC,DELTA0); WRITELN;
END;
WRITELN;
WRITELN (' l,b,r: heliocentric ecliptic (geometric) ');
WRITE (' RA,Dec: geocentric equatorial ');
IF MODE IN ['A','a'] THEN WRITELN('(apparent)')
ELSE WRITELN('(astrometric)');
WRITELN (' delta: geocentric distance (geometric)');
WRITELN;
END;

UNTIL MODE IN ['E','e']

END. (* PLANPOS *)
(*-----*)

```

To begin with, PLANPOS asks for the equinox to which the positions to be calculated should be referred. Possible entries are A (apparent coordinates, to the equinox of date), J (J2000, astrometric) and B (B1950, astrometric). This choice is always redisplayed after the coordinates have been computed, not just at the beginning of the program. It is always possible to exit from the program at this point by entering (E).

In the following example, we first choose apparent coordinates. After the appropriate response, we enter the exact time for which the coordinates are to be calculated. Note that the time must be given in Ephemeris Time (ET)! In the example, we select 1989 January 1 at 00:00 ET. (As usual, all data entries are shown in *italic*.) PLANPOS now returns the desired coordinates.

PLANPOS: geocentric and heliocentric planetary positions

version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

(J) J2000 astrometric (B) B1950 astrometric

(A) apparent coordinates (E) end

enter option: *A*

date (year month day hour) ? *1989 1 1 0.0*

date: 1989/ 1/ 1 0.0(ET) JD: 2447527.500 equinox of date

	l o ' "			b o ' "			r AU	RA h m s			Dec o ' "			delta AU
Sun	0	0	0.0	0	0	0.0	0.000000	18	45	53.71	-23	1	25.5	0.983309
Mercury	347	56	35.7	-6	5	21.5	0.370100	19	59	16.66	-22	34	11.9	1.175632
Venus	226	5	15.0	1	43	26.9	0.723666	17	7	15.24	-22	3	57.5	1.522289
Earth	100	33	9.4	0	0	0.2	0.983309	0	0	0.00	0	0	0.0	0.000000
Mars	60	32	51.9	0	21	19.3	1.49745	1	13	47.46	8	24	5.3	0.97649
Jupiter	64	30	7.8	0-45	53.2		5.03195	3	38	35.12	18	33	7.5	4.27637
Saturn	275	6	53.5	0	47	14.5	10.04354	18	24	14.55	-22	36	28.2	11.02274
Uranus	271	18	42.6	0-13	47.8		19.31483	18	7	39.26	-23	39	0.8	20.28599
Neptune	279	54	35.7	0	55	55.8	30.21917	18	42	54.14	-22	10	16.6	31.20230
Pluto	222	55	15.8	15	52	34.5	29.65861	15	6	35.07	-1	16	18.2	30.17673

l,b,r: heliocentric ecliptic (geometric)

RA,Dec: geocentric equatorial (apparent)

delta: geocentric distance (geometric)

It should be noted that the heliocentric ecliptic coordinates of the Sun, and the geocentric equatorial coordinates of the Earth are always set to zero.

For comparison purposes, we now want to calculate the positions of the planets for the same time, but referred to equinox J2000. The small differences that result are caused by precession over a period of 11 years.

(J) J2000 astrometric (B) B1950 astrometric
 (A) apparent coordinates (E) end

enter option: *J*

date (year month day hour) ? 1989 1 1 0.0

date: 1989/ 1/ 1 0.0(ET) JD: 2447527.500 equinox J2000

	l			b			r	RA			Dec			delta
	o	'	"	o	'	"	AU	h	m	s	o	'	"	AU
Sun	0	0	0.0	0	0	0.0	0.000000	18	46	34.57	-23	0	32.4	0.983309
Mercury	348	5	49.4	-6	5	22.1	0.370100	19	59	56.55	-22	32	12.3	1.175632
Venus	226	14	28.3	1	43	22.9	0.723666	17	7	55.77	-22	4	40.7	1.522289
Earth	100	42	22.5	0	0	5.2	0.983309	0	0	0.00	0	0	0.0	0.000000
Mars	60	42	5.0	0	21	24.1	1.49745	1	14	21.40	8	27	27.8	0.97649
Jupiter	64	39	21.0	0-45	48.3		5.03195	3	39	11.60	18	35	3.6	4.27637
Saturn	275	16	6.7	0	47	9.5	10.04354	18	24	55.40	-22	35	56.0	11.02274
Uranus	271	27	55.8	0-13	52.9		19.31483	18	8	20.47	-23	38	44.9	20.28599
Neptune	280	3	48.9	0	55	50.8	30.21917	18	43	34.77	-22	9	26.4	31.20230
Pluto	223	4	29.9	15	52	30.6	29.65861	15	7	9.55	-1	18	40.7	30.17673

l,b,r: heliocentric ecliptic (geometric)

RA,Dec: geocentric equatorial (apparent)

delta: geocentric distance (geometric)

It will be seen that the differences are very small, but still to be taken into account. In comparing such results with yearbooks some care should be taken. Sometimes only the positions of the planets from Mercury to Neptune are given in apparent coordinates, while the ephemeris for Pluto uses astrometric coordinates. The reason for this is that in observing this faint planet it is advisable to use an atlas or a finder chart, in order to be sure of identifying it.

(J) J2000 astrometric (B) B1950 astrometric
 (S) apparent coordinates (E) end

enter option: *E*

After the next response *E* the program terminates.

7. Physical Ephemerides of the Planets

In the previous chapter we have provided the means of accurately calculating the positions of the major planets. Anyone who frequently observes the planets, the Sun, or the Moon, is also interested in data that describe the aspect of the various bodies. The visual magnitude and the apparent angular diameter are data of fundamental importance. Anyone who wants to observe and plot surface markings or atmospheric structure also needs to know the orientation of the planet's axis of rotation (and thus of the planetographic coordinate system), relative to the line of sight, together with the conditions of illumination. We intend to show how these data may be calculated. In doing so, we shall try to achieve an accuracy that should suffice, in most instances, for practical observations.

The data on the size, form and rotation of the planets that are required for calculation are generally taken from the *Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements* of 1982, which is listed in the Bibliography. The physical ephemerides calculated using these data are consistent with the values currently given in the *Astronomical Almanac*.

Because the physical ephemerides depend on the relative position of the Sun, the Earth, and a planet, we require the heliocentric coordinates of the Earth and the planet. The planetary coordinates that we calculated in our discussion of the two-body problem have sufficient accuracy for practical observational purposes. An appropriate subroutine (**POSITION**) has already been described in Sect. 5.2. When necessary, we can, of course, also fall back on the routines discussed in Chap. 6 for accurate planetary coordinates. The method of determining physical ephemerides remains the same.

7.1 Rotation

Just as the rotation of the Earth is determined by the direction of the North Celestial Pole and sidereal time, the rotation of the other planets may be described by the position of the corresponding north poles and rotation through a specific angle. Following a recommendation of the International Astronomical Union (IAU) the north pole is defined as the pole of a planet, minor planet, or satellite, that lies on the northern side of what is known as the *invariable plane* of the Solar System. This plane is normal to the Solar System's combined angular momentum vector, but for simplicity it may be visualized

Table 7.1. Orientation of the rotational axes of the Sun and planets

	α_0 (J2000)	δ_0 (J2000)
Sun	285°96	+63°96
Mercury	281°02 -0°033T	+61°45 -0°005T
Venus	272°78	+67°21
Earth	0°00 -0°641T	+90°00 -0°557T
Mars	317°681-0°108T	+52°886-0°061T
Jupiter	268°05 -0°009T	+64°49 +0°003T
Saturn	40°66 -0°036T	+83°52 -0°004T
Uranus	257°43	-15°10
Neptune	295°33	+40°65
Pluto	311°63	+4°18

as the mean orbital plane of all the planets. Because the orbits of the major planets are only slightly inclined to the Earth's orbital plane, the invariable plane is very nearly identical with the ecliptic.

According to the definition just given, not all the planets rotate like the Earth. The exceptions are Venus, Uranus, and Pluto, which rotate clockwise when looking down on the north pole. Unlike the *direct rotation* of most of the planets, their rotation is *retrograde*.

As well as calculating the rotational phase of the major planets, we also want to determine similar data for the Sun. In what follows, when we discuss planets, it should be borne in mind that the procedure described for determining the rotational phase may also be applied to the Sun. Factors specific to that particular body are discussed in a separate section.

Table 7.1 gives a summary of the orientations of the rotational axes of the Sun and planets. The columns show the right ascension α_0 and declination δ_0 of the north pole, referred to the J2000 equinox. The argument

$$T = (\text{JD} - 2451545)/36525 \quad (7.1)$$

indicates, as usual, the number of Julian centuries since the J2000 epoch.

7.1.1 The Position Angle of the Axis

The position of the rotational axis of a planet, as seen by an observer, is usually described by giving the position angle of the axis and the latitude of the Earth relative to the planet's equator (see Fig. 7.1). The position angle ϑ of the planet's axis is taken to be the angle between the projection of the rotational axis onto the celestial sphere and the observer's North. The position angle is measured from North ($\vartheta = 0^\circ$) through East ($\vartheta = 90^\circ$).

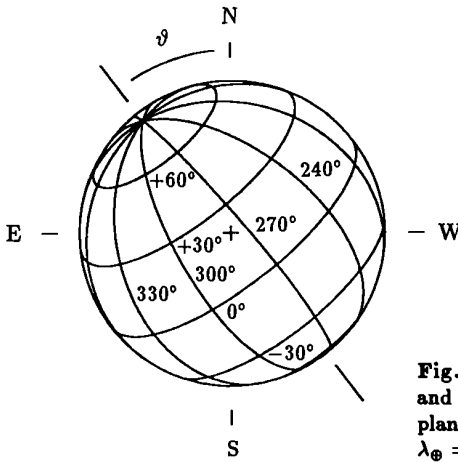


Fig. 7.1. Position angle of the rotational axis and planetographic coordinates (aspect of the planet Mars on 1993 Oct. 01, 0^h ET; $\vartheta = 37^\circ.9$, $\lambda_\oplus = 276^\circ.1$, $\varphi_\oplus = 20^\circ.4$)

To calculate the position angle of the planet's axis we begin with the geocentric equatorial coordinates $\mathbf{r} = (x, y, z)$ of the planet. In addition, let

$$\mathbf{d} = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = \begin{pmatrix} \cos \alpha_0 \cos \delta_0 \\ \sin \alpha_0 \cos \delta_0 \\ \sin \delta_0 \end{pmatrix} \quad (7.2)$$

be the vector, taken from the centre of the planet, describing the direction of the axis. With the aid of the vector $\mathbf{r}$, we now form the three, mutually perpendicular, unit vectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, so that $\mathbf{e}_1$ is directed towards the planet, $\mathbf{e}_2$ is directed towards increasing right ascension (i.e., towards East), and $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$ is directed towards increasing declination (towards North) as shown in Fig. 7.2. With this construction, we have

$$\mathbf{e}_1 = \frac{1}{r} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \mathbf{e}_2 = \frac{1}{\varrho} \begin{pmatrix} -y \\ +x \\ 0 \end{pmatrix} \quad \mathbf{e}_3 = \frac{1}{r\varrho} \begin{pmatrix} -xz \\ -yz \\ x^2 + y^2 \end{pmatrix}, \quad (7.3)$$

whence

$$r = \sqrt{x^2 + y^2 + z^2} \quad \text{and} \quad \varrho = \sqrt{x^2 + y^2}$$

represent the value of the radius vector and its projection onto the plane of the equator. The position angle ϑ , at which we see the direction vector $\mathbf{d}$, is measured in the plane defined by $\mathbf{e}_2$ and $\mathbf{e}_3$, which is normal to the line

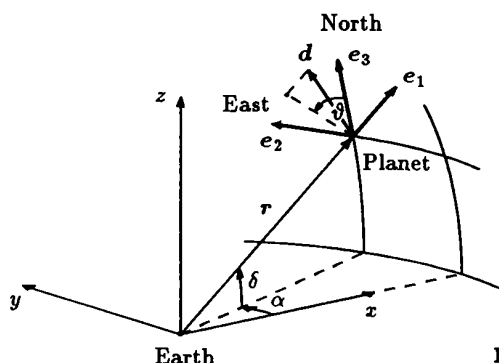


Fig. 7.2. Calculating the position angle

of sight. By projecting the direction vector d onto e_2 and e_3 we obtain the following expressions for determining the position angle ϑ :

$$\begin{aligned}\cos \vartheta &= e_3 \cdot d = ((-xz)d_x + (-yz)d_y + (x^2 + y^2)d_z)/(r\rho) \\ \sin \vartheta &= e_2 \cdot d = ((-y)d_x + (x)d_y)/\rho.\end{aligned}\quad (7.4)$$

The POSANG function utilizes these equations to calculate the position angle:

```
(*-----*)
(* POSANG: computes the position angle from given directions *)
(* X,Y,Z      Coordinates of the planet relative to the observer *)
(* DX,DY,DZ   Direction vector *)
(* POSANG     Position angle (0<=POSANG<360deg) *)
(* Both vectors must be given in a common coordinate system *)
(* (e.g. mean equator and equinox of date) *)
(*-----*)
FUNCTION POSANG ( X,Y,Z, DX,DY,DZ : REAL ) : REAL;
  VAR C,S,PHI : REAL;
  BEGIN
    C := ( (-X*Z)*DX + (-Y*Z)*DY + (X*X+Y*Y)*DZ ) / SQRT (X*X+Y*Y+Z*Z);
    S := ( (-Y)*DX + (X)*DY );
    PHI := ATN2 ( S, C );
    IF (PHI<0.0) THEN POSANG:=PHI+360.0 ELSE POSANG:=PHI
  END;
(*-----*)
```

The POSANG subroutine may also be used to calculate the position angle of the Sun, if we substitute the planetocentric coordinates of the Sun instead of the direction vector d . This will be discussed in more detail in Sect. 7.2 in connection with the illumination geometry of the planets.

7.1.2 Planetographic Coordinates

Like the geographical coordinate system used on Earth, we may apply a system of spherical coordinates, fixed with reference to the surface, to each of the other planets. This may then be used for position measurements. The equator of the respective body serves as the reference plane for the system. Because most of the planets in the Solar System are flattened as a result of their rotation, a distinction is drawn between two definitions of latitude:

- The *planetocentric* latitude φ' indicates the angle between the direction vector of a point and the planet's equator. It therefore corresponds to declination in the geocentric equatorial coordinate system.
- The *planetographic* latitude φ differs from the planetocentric latitude φ' when the planet is flattened. If from a given point, we use a plumb-line to drop a normal to the surface of a planet, the planetographic latitude is the angle between this line and the equatorial plane. In the case of the Earth, it thus corresponds to geographical latitude (see Fig. 9.5).

The relationship between geocentric latitude φ' and geographic latitude φ is described in Sect. 9.3. We simply need to substitute the equatorial radius R_{Eq} of the planet for the Earth's radius $R_{\oplus}$, whence the flattening of the planetary ellipsoid is defined as

$$f = \frac{R_{\text{Eq}} - R_{\text{Pol}}}{R_{\text{Eq}}} \quad (7.5)$$

Conversion between planetocentric latitude φ' and planetographic latitude φ may be made by using the equation

$$\tan \varphi' = (1 - f)^2 \tan \varphi \quad (7.6)$$

To calculate physical ephemerides we set the appropriate values for the equatorial radius and flattening of the major planets using the SHAPE procedure, so that the data are independent of the calculation routines.

```
(*-----*)
(* SHAPE: returns the equatorial radii and flattening of planets *)
(* PLANET Name of the planet *)
(* R_EQU Equatorial radius (km) *)
(* FL Geometric flattening *)
(*-----*)
```

```
PROCEDURE SHAPE ( PLANET: PLANET_TYPE; VAR R_EQU, FL: REAL );
```

```
  BEGIN
```

```
    CASE PLANET OF
```

```
      MERCURY: BEGIN R_EQU := 2439.0; FL := 0.0      END;
      VENUS : BEGIN R_EQU := 6051.0; FL := 0.0      END;
      EARTH : BEGIN R_EQU := 6378.14; FL := 0.00335281 END;
      MARS  : BEGIN R_EQU := 3393.4; FL := 0.0051865  END;
      JUPITER: BEGIN R_EQU := 71398.0; FL := 0.0648088 END;
```

```

SATURN : BEGIN R_EQU := 60000.0; FL := 0.1076209 END;
URANUS : BEGIN R_EQU := 25400.0; FL := 0.030 END;
NEPTUNE: BEGIN R_EQU := 24300.0; FL := 0.0259 END;
PLUTO : BEGIN R_EQU := 1500.0; FL := 0.0 END;
END;
END;
(*-----*)

```

Here the previously defined type

```

TYPE PLANET_TYPE = ( MERCURY, VENUS, EARTH, MARS,
                     JUPITER, SATURN, URANUS, NEPTUNE, PLUTO );

```

is employed to designate the individual planet.

As with the determination of the latitude of a point, we differentiate between planetocentric and planetographic longitude:

- Planetocentric longitude λ' is measured positively towards the east, irrespective of the direction of rotation.
- Planetographic longitude λ is reckoned in the opposite sense to the rotation. Because of this definition, the longitude of what is known as the *central meridian*, which runs through the centre of the disk as seen from Earth, always increases with increasing time.

For the planets Venus, Uranus, and Pluto, which have retrograde rotations, both longitude definitions are identical ($\lambda = \lambda'$). On the other hand, the planetocentric and planetographic longitudes differ in sign ($\lambda = -\lambda'$) for the planets Mercury, Mars, Jupiter, Saturn and Neptune, which, like the Earth, rotate from West to East. In the case of the Sun, however, it should be noted that what is known as heliographic longitude increases in the direction of rotation, and is thus identical with heliocentric longitude, despite the fact that the Sun has direct rotation.

Establishment of a reference point (the prime meridian) for longitude measurements either makes use of prominent features on the surface, or, failing these, is defined in an appropriate manner. With the giant planets Jupiter, Saturn, Uranus, and Neptune, which do not rotate like solid bodies with a rigid surface, rotation systems with constant angular velocities have been introduced. For Jupiter's atmosphere, which rotates with a period of $9^{\text{h}}30^{\text{m}}$ at the equator, and $9^{\text{h}}56^{\text{m}}$ at higher latitudes, System I applies to regions near the equator ($|\varphi| \leq 9^\circ$), and System II for all higher latitudes. Radio-astronomical studies of the outer planets have shown that these planets' magnetic fields rotate with periods that do not agree with the optically determined periods for System I or II. The reason for this is that the basic causes of the radio emission lie in the planetary interiors, which essentially rotate as solid bodies, quite independently of the atmospheric layers. System III has therefore been introduced to describe the rotation relative to the radio emission. For Uranus and Neptune only System III is used nowadays.

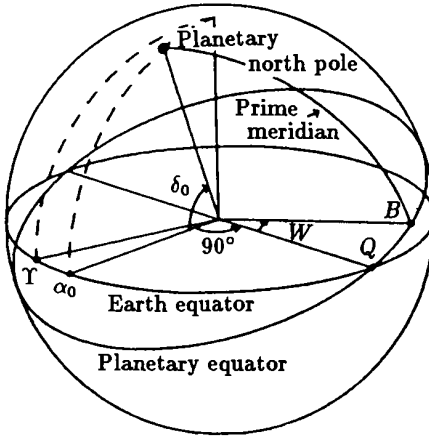


Fig. 7.3. Orientation of the prime meridian

For Saturn, in addition to System III, we also give details for System I, so that comparisons with older drawings will still be possible.

We use the angle W to describe the position of a planet's prime meridian. It is measured

- from the intersection of the planes of the Earth's equator (J2000) and the planet's equator (point Q)
- along the planet's equator
- increasing in the direction of rotation
- to the intersection of the prime meridian and the planet's equator (point B).

Of the two points at which the planes of the Earth's and the planet's equators intersect, point Q is thus defined as the one at which a co-rotating point on the planet's equator crosses the plane of the Earth's equator from south to north (see Fig. 7.3). The angle W increases linearly with time and may be calculated from the data given in Table 7.2. The term

$$d = \text{JD} - 2451545 \quad (7.7)$$

indicates the number of days that have elapsed since the standard epoch J2000.

The numerical values of the parameters α_0 , δ_0 and W , used to describe the position of the rotational axis and the prime meridian, are brought together in the **ORIENT** procedure. As a result, any future changes in the rotational parameters may be taken into account very easily. As well as the time and

Table 7.2. Orientation of the prime meridian of the Sun and planets

		<i>W</i> (J2000)	
Sun		84°11	+ 14°1844000 <i>d</i>
Mercury		329°71	+ 6°1385025 <i>d</i>
Venus		159°91	- 1°4814205 <i>d</i>
Earth		100°21	+ 360°9856123 <i>d</i>
Mars		176°655	+ 350°8919830 <i>d</i>
Jupiter	System I	67°1	+ 877°900 <i>d</i>
	System II	43°3	+ 870°270 <i>d</i>
	System III	284°95	+ 870°536 <i>d</i>
Saturn	System I	227°2037	+ 844°300 <i>d</i>
	System III	38°90	+ 810°7939024 <i>d</i>
Uranus	System III	261°62	- 554°913 <i>d</i>
Neptune	System III	107°21	+ 468°75 <i>d</i>
Pluto		252°66	- 56°364 <i>d</i>

the name of the relevant planet, ORIENT requires the rotation system to be entered. For this we use the variable SYSTEM which is of type

```
TYPE SYSTEM_TYPE = ( SYS_I, SYS_II, SYS_III );
```

Depending on the value of this parameter, and for the planets Jupiter and Saturn, ORIENT will return (in *W*) the position of the prime meridian for the equatorial regions, for higher latitudes, or for the radio emission.

```
(*-----*)
(*)
(* ORIENT: returns the elements describing the planetocentric *)
(* coordinate system of a planet *)
(*)
(*) PLANET Name of the planet *)
(*) SYSTEM System of rotation (I, II or III) *)
(*) T Time in Julian centuries since J2000 (ET or TDB/TDT) *)
(*) A Right ascension of the axis of rotation *)
(*) D Declination of the axis of rotation *)
(*) W Orientation of the prime meridian with respect to the *)
(*) intersection of the Earth's equator (J2000) and the *)
(*) planetary equator of date *)
(*) SENSE Sense of rotation (DIRECT or RETROGRADE) *)
(*)
(*-----*)
```

```
PROCEDURE ORIENT ( PLANET: PLANET_TYPE; SYSTEM: SYSTEM_TYPE; T: REAL;
VAR A,D,W: REAL; VAR SENSE: ROTATION_TYPE );
```

```
VAR TD : REAL;
```

```
BEGIN
```

```
  (* Compute right ascension and declination of the axis of *)
  (* rotation with respect to the equator and equinox of J2000 *)
  CASE PLANET OF
```

```

MERCURY : BEGIN A := 281.02 - 0.033*T; D := 61.45 -0.005*T END;
VENUS   : BEGIN A := 272.78           ; D := 67.21           END;
EARTH   : BEGIN A := 0.00 - 0.641*T; D := 90.00 -0.557*T END;
MARS    : BEGIN A := 317.681- 0.108*T; D := 52.886-0.061*T END;
JUPITER : BEGIN A := 268.05 - 0.009*T; D := 64.49 +0.003*T END;
SATURN  : BEGIN A := 40.66 - 0.036*T; D := 83.52 -0.004*T END;
URANUS  : BEGIN A := 257.43           ; D := -15.10           END;
NEPTUNE : BEGIN A := 295.33           ; D := 40.65           END;
PLUTO   : BEGIN A := 311.63           ; D := 4.18             END;

END;
(* Compute orientation of the prime meridian *)
TD := 36525.0 * T;
CASE PLANET OF
  MERCURY : W := 329.710 + 6.1385025*TD;
  VENUS   : W := 159.910 - 1.4814205*TD;
  EARTH   : W := 100.21 + 360.9856123*TD;
  MARS    : W := 176.655 + 350.8919830*TD;
  JUPITER : CASE SYSTEM OF
    SYS_I   : W := 67.10 + 677.900*TD;
    SYS_II  : W := 43.30 + 870.270*TD;
    SYS_III : W := 284.95 + 670.536*TD
  END;
  SATURN  : CASE SYSTEM OF
    SYS_I, SYS_II : W := 227.2037 + 844.3000000*TD;
    SYS_III       : W := 38.90 + 810.7939024*TD
  END;
  URANUS  : W := 261.620 - 554.913*TD; (* System III *)
  NEPTUNE : W := 107.210 + 468.75 *TD; (* System III *)
  PLUTO   : W := 252.660 - 56.364*TD
END;
W := W/360.0; W := 360.0*(W-TRUNC(W));
(* Define sense of rotation *)
CASE PLANET OF
  MERCURY,EARTH,MARS,
  JUPITER,SATURN,NEPTUNE : SENSE := DIRECT;
  VENUS,URANUS,PLUTO     : SENSE := RETROGRADE;
END;
END;
(*-----*)

```

If we know the position of a planet's rotational axis and prime meridian, we can determine the planetocentric and planetographic coordinates of the Sun and Earth, which are of particular importance for observation of a planet. These values describe, for example, how far the planet's north pole is inclined towards the Earth; which point of the planet's surface is at the centre of the apparent disk; and which point is vertically illuminated by the Sun.

To calculate the planetocentric longitude and latitude of the Earth or the Sun, we employ a fixed coordinate system that rotates with the planet and that is specified by the three, mutually perpendicular, unit vectors (e_1, e_2, e_3). The vector e_3 lies parallel to the rotational axis, while e_1 is directed from the centre of the planet to the intersection of the equator and the prime meridian.

Expressed in full, the components of these vectors, in equatorial coordinates, are

$$\begin{aligned}
 \mathbf{e}_1 &= \begin{pmatrix} -\cos W \sin \alpha_0 - \sin W \sin \delta_0 \cos \alpha_0 \\ +\cos W \cos \alpha_0 - \sin W \sin \delta_0 \sin \alpha_0 \\ +\sin W \cos \delta_0 \end{pmatrix} \\
 \mathbf{e}_2 &= \begin{pmatrix} +\sin W \sin \alpha_0 - \cos W \sin \delta_0 \cos \alpha_0 \\ -\sin W \cos \alpha_0 - \cos W \sin \delta_0 \sin \alpha_0 \\ +\cos W \cos \delta_0 \end{pmatrix} \\
 \mathbf{e}_3 &= \begin{pmatrix} +\cos \delta_0 \cos \alpha_0 \\ +\cos \delta_0 \sin \alpha_0 \\ +\sin \delta_0 \end{pmatrix} .
 \end{aligned} \tag{7.8}$$

These expressions are easily obtained from the known relationships between the Gaussian vectors $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ (see Sect. 4.5), if we replace the orbital elements i , Ω and ω , respectively, by: the angle $90^\circ - \delta_0$ between the Earth's and the planet's equator; the right ascension $\alpha_0 + 90^\circ$ of the line of intersection of the Earth's and the planet's equator; and the angle W between this line of intersection and the prime meridian. Similarly, to convert equatorial coordinates into the planet's fixed coordinate system, the matrix $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ may be calculated by a call to the GAUSVEC routine in the form of

GAUSVEC (90.0+A, 90.0-D, W, E) .

If we represent the geocentric planetary coordinates by $\mathbf{r} = (x, y, z)$, then, by projecting $-\mathbf{r}$ onto the direction vectors $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$, we obtain the Cartesian planetocentric coordinates of the Earth

$$\begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} = \begin{pmatrix} -\mathbf{e}_1 \cdot \mathbf{r} \\ -\mathbf{e}_2 \cdot \mathbf{r} \\ -\mathbf{e}_3 \cdot \mathbf{r} \end{pmatrix} = \begin{pmatrix} r \cos \varphi' \cos \lambda' \\ r \cos \varphi' \sin \lambda' \\ r \sin \varphi' \end{pmatrix} . \tag{7.9}$$

from which, by using

$$\begin{aligned}
 \tan \varphi' &= \frac{s_z}{\sqrt{s_x^2 + s_y^2}} \\
 \tan \lambda' &= \frac{s_y}{s_x} \\
 \tan \varphi &= \frac{\tan \varphi'}{(1-f)^2} \\
 \lambda &= \begin{cases} +\lambda' & (\text{retrograde rotation}) \\ -\lambda' & (\text{direct rotation}) \end{cases}
 \end{aligned} \tag{7.10}$$

we may determine the corresponding values for the planetocentric and planetographic longitude and latitude of the Earth.

The planetocentric and planetographic coordinates of the Sun may be calculated in precisely the same way. We simply need to substitute the heliocentric planetary coordinates for the geocentric planetary coordinates.

The steps required to calculate the various rotational parameters of a planet that have been mentioned are all combined in the ROTATION subroutine. Apart from planetographic longitude and latitude, which are normally given in yearbooks, the routine also returns the planetocentric latitude, which will be required later when we come to calculate the illumination. The direction vector of the rotational axis, which is also calculated, additionally serves to determine the position angle of the axis by means of the POSANG function. If the heliocentric coordinates of the planet are entered instead of the geocentric ones, we obtain the planetographic coordinates of the Sun.

```
(*-----*)
(* ROTATION: computes the rotation parameters of a planet *)
(* X,Y,Z   Geocentric equatorial coordinates of the planet (AU) *)
(* A       Right ascension of the axis of rotation (J2000) *)
(* D       Declination of the axis of rotation (J2000) *)
(* W       Orientation of the prime meridian with respect to the *)
(*          intersection of the Earth's equator (J2000) and the *)
(*          planetary equator of date *)
(* SENSE    Sense of rotation (DIRECT or RETROGRADE) *)
(* FLATT    Geometric flattening of the planet *)
(* AX,AY,AZ Rotation axis unit vector (J2000) *)
(* LONG     Planetographic longitude of the Earth (deg) *)
(* LAT      Planetographic latitude of the Earth (deg) *)
(* DEC      Planetocentric latitude of the Earth (deg) *)
(* The heliocentric coordinates of the planet may be substituted for *)
(* X,Y,Z to obtain the planetographic coordinates of the Sun. *)
(*-----*)
```

```
PROCEDURE ROTATION (X,Y,Z, A,D,W: REAL; SENSE: ROTATION_TYPE; FLATT: REAL;
VAR AX,AY,AZ, LONG,LAT,DEC: REAL );
```

```
VAR SX,SY,SZ,T: REAL;
    E : REAL33;
```

```
BEGIN
```

```
(* Compute unit vectors E(*,1) (intersection of prime meridian and *)
(* planetary equator), E(*,3) (parallel to the rotation axis) and *)
(* E(*,2) perpendicular to E(*,1) and E(*,3)) *)
GAUSVEC ( 90.0+A,90.0-D,W, E );
```

```
(* Copy rotation axis unit vector *)
AX := E[1,3]; AY := E[2,3]; AZ := E[3,3];
```

```
(* Compute planetocentric latitude and longitude *)
SX := - ( E[1,1]*X + E[2,1]*Y + E[3,1]*Z );
SY := - ( E[1,2]*X + E[2,2]*Y + E[3,2]*Z );
SZ := - ( E[1,3]*X + E[2,3]*Y + E[3,3]*Z );
```

```

T := SZ / SQRT(SI*SI+SY*SY);
DEC := ATN ( T );
LONG := ATN2 ( SY, SI );

(* Compute planetographic latitude and longitude *)
IF (SENSE=DIRECT) THEN LONG:=-LONG;
IF LONG<0.0 THEN LONG:=LONG+360.0;
LAT := ATN ( T / ((1.0-FLATT)*(1.0-FLATT)) );

END;

```

(*-----*)

7.2 Illumination Conditions

Apart from the description of the actual rotational phase, planetary observers are also interested in the conditions of illumination. These are described by the phase, the position angle of the Sun, and the planet's magnitude.

7.2.1 Phase and Elongation

Because the planets in the Solar System do not shine with their own light, but are illuminated by the Sun, generally only a portion of the planet's disk visible from the Earth is illuminated. A measure of the illuminated portion is given by the angle between the planetocentric direction vectors of the Sun and Earth, which is known as the *phase angle* i . A phase angle of $i = 0^\circ$ corresponds to a fully illuminated planetary disk, and one of $i = 90^\circ$ to half phase.

In the Sun-Earth-Planet triangle (Fig. 7.4) let the distance of the planet from the Sun be r , that of the Earth R , and the distance between the planet

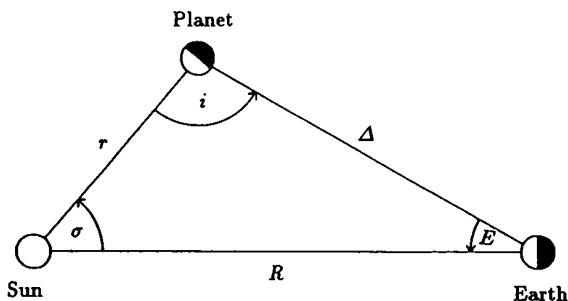


Fig. 7.4. Phase angle i and elongation E

and the Earth Δ . The phase angle may be obtained from the cosine law

$$\cos i = \frac{\Delta^2 + r^2 - R^2}{2r\Delta} \quad (7.11)$$

The illuminated fraction of the planetary disk, known as the *phase* is obtained from

$$k = \frac{1 + \cos i}{2} \quad (7.12)$$

Note that k does not apply to the spherical surface of the planet, but to the surface of the planet's apparent disk.

As with the phase angle, the cosine law may also be applied to the Sun-Earth-Planet triangle to give us

$$\cos E = \frac{\Delta^2 + R^2 - r^2}{2R\Delta} \quad (7.13)$$

where E is the *elongation* of the Sun, i.e., the angular distance between the Sun and the planet as seen from Earth. The elongation is an easy way of estimating whether the planet can be observed, or whether it is too close to the Sun in the sky to be visible.

The equations required to calculate the elongation and phase of a planet are combined in the ILLUM routine. The values to be entered are the heliocentric coordinates of the planet and the Earth. They must be given in a common coordinate system and for the same equinox.

```
(*-----*)
(* ILLUM: Computes the illumination parameters of a planet *)
(* X,Y,Z      Heliocentric coordinates of the planet *)
(* IE,YE,ZE   Heliocentric coordinates of the Earth *)
(* R          Heliocentric distance of the planet *)
(* D          Geocentric distance of the planet *)
(* ELONG      Elongation (deg) *)
(* PHI        Phase angle (deg) *)
(* K          Phase *)
(* Note: All coordinates must refer to the same coordinate system. *)
(*-----*)
PROCEDURE ILLUM ( X,Y,Z, IE,YE,ZE: REAL; VAR R,D,ELONG,PHI,K: REAL );
  VAR XP,YP,ZP, RE, C_PHI : REAL;
  BEGIN
    (* Compute the planet's geocentric position *)
    XP:=X-IE; YP:=Y-YE; ZP:=Z-ZE;
    (* Compute the distances in the Sun-Earth-planet triangle *)
    R := SQRT ( X*X + Y*Y + Z*Z ); (* Sun-planet distance *)
    RE := SQRT ( IE*IE + YE*YE + ZE*ZE ); (* Sun-Earth distance *)
    D := SQRT ( XP*XP + YP*YP + ZP*ZP ); (* Earth-planet distance *)
    (* Compute elongation, phase angle and phase *)
    ELONG := ACS ( ( D*D + RE*RE - R*R ) / ( 2.0*D*RE ) );
    C_PHI := ( D*D + R*R - RE*RE ) / ( 2.0*D*R );
    PHI := ACS ( C_PHI );
    K := 0.5*(1.0+C_PHI);
  END;
(*-----*)
```

7.2.2 The Position Angle of the Sun

Because the planets do not orbit precisely in the plane of the ecliptic, it is rare for sunlight to fall on their surfaces from exactly (our) east or west. In general, therefore, the dividing line between the illuminated and dark hemispheres of a planet (the terminator) does not pass through the two poles of the apparent disk.

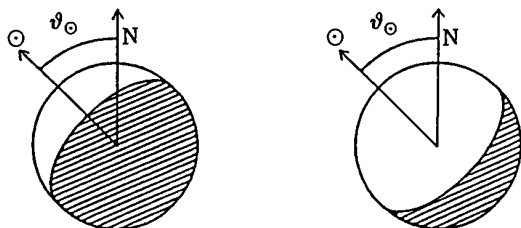


Fig. 7.5. Position angle of the Sun at various phase angles

The orientation of the illuminated phase relative to the observer's north is unambiguously described by the position angle $\vartheta_{\odot}$ of the Sun, which is occasionally described as the illumination angle (Fig. 7.5). Using the previously determined relationship (7.4) for the position angle of the rotational axis, we can also calculate the position angle of the Sun. Instead of the direction vector of the axis, we substitute the direction vector directed from the planet towards the Sun.

We can therefore employ the POSANG procedure to calculate the illumination angle. We thus enter the geocentric equatorial coordinates of the planet for vector $\mathbf{r}$ and the planetocentric equatorial coordinates of the Sun for the direction vector $\mathbf{d}$. These may be obtained by altering the sign of the heliocentric planetary coordinates that are obtained in any case during the calculation. Because position angles are normally referred to the actual direction of North, the solar and planetary coordinates should also be referred to the equinox of date.

7.2.3 Apparent Magnitude

By the apparent magnitude of a planet we mean the total brightness, expressed in magnitudes, of the illuminated portion visible from Earth. It depends on the planet's distance from the Sun and from the Earth, and also on the size and composition of its surface or atmosphere. Consequently, the apparent magnitude $V(1,0)$, of a planet at a distance of $r = \Delta = 1$ AU from the Sun and the Earth, and at a phase angle of $i = 0^\circ$, would, in accordance with its diameter and reflectivity, range between magnitude $-0^m.4$ (Mercury) and $-9^m.4$ (Jupiter).

Table 7.3. Apparent magnitudes of the planets

Planet	$V(1,0)$	$\Delta m(i)$
Mercury	$-0^m.42$	$+ 3^m.80 \cdot \left(\frac{i}{100^\circ}\right) - 2^m.73 \cdot \left(\frac{i}{100^\circ}\right)^2 + 2^m.00 \cdot \left(\frac{i}{100^\circ}\right)^3$
Venus	$-4^m.40$	$+ 0^m.09 \cdot \left(\frac{i}{100^\circ}\right) + 2^m.39 \cdot \left(\frac{i}{100^\circ}\right)^2 - 0^m.65 \cdot \left(\frac{i}{100^\circ}\right)^3$
Earth	$-3^m.86$	
Mars	$-1^m.52$	$+ 1^m.60 \cdot \left(\frac{i}{100^\circ}\right)$
Jupiter	$-9^m.40$	$+ 0^m.50 \cdot \left(\frac{i}{100^\circ}\right)$
Saturn	$-8^m.88$	$- 2^m.60 \cdot \sin \varphi'_\odot + 1^m.25 \cdot \sin \varphi'_\odot ^2 + 4^m.40 \cdot \lambda'_\oplus - \lambda'_\odot $
Uranus	$-7^m.19$	
Neptune	$-6^m.87$	
Pluto	$-1^m.0$	

The strength of the illumination of a planet is inversely proportional to the square of its distance from the Sun. Similarly, the observed radiation flux for a planet at a given distance from the Sun decreases as the square of its distance from the Earth. Bearing in mind that a decrease in the radiation flux by a factor of 100 corresponds to a change of +5 magnitudes in the brightness, we find that the apparent brightness of a planet is given by

$$m = V(1,0) + 5 \log \left(\frac{r \cdot \Delta}{\text{AU}^2} \right) + \Delta m(i) \quad (7.14)$$

Here, the $\Delta m(i)$ term represents the magnitude's phase-dependence. This involves both the decrease in illuminated area with decreasing phase, and also the dependence of scattering and reflection upon the angle of illumination. In general, $\Delta m(i)$ is expressed in the form of a power law, determined from observational data.

In the case of Saturn, the ring system's contribution to the planet's brightness also needs to be taken into account. This may be described as a function of the planetocentric latitude $\varphi'_\odot$ of the Sun, and the difference in the planetocentric longitudes of the Sun and the Earth, $\lambda'_\oplus - \lambda'_\odot$.

Expressions for the absolute magnitude $V(1,0)$ and the phase-dependent factor $\Delta m(i)$ for the planets are given in Table 7.3. The function BRIGHT uses these values to determine the apparent magnitudes of the planets.

```

(*-----*)
(*)
(* BRIGHT: Computes the apparent magnitude of a planet *)
(*)
(* PLANET      Name of the planet *)
(* R           Heliocentric distance of the planet (AU) *)
(* DELTA       Distance of the planet from the observer (AU) *)
(* PHI         Phase angle (deg) *)
(* DEC         Planetocentric latitude of the observer (deg) *)
(* DLONG       Difference of the planetocentric longitudes of the Sun *)

```

```

(*)           and the observer (deg)           (*)
(*)
(*) Magnitudes V(1,0) from Astronomical Almanac 1984. DEC and DLONG are (*)
(*) only required to compute the apparent brightness of Saturn, which (*)
(*) depends on the ring orientation.          (*)
(*)                                           (*)
(*)-----*)

```

```
FUNCTION BRIGHT ( PLANET: PLANET_TYPE; R,DELTA,PHI,DEC,DLONG: REAL ): REAL;
```

```
CONST LN10 = 2.302585093;    (* Natural logarithm of 10 *)
```

```
VAR P,SD,DL,MAG : REAL;
```

```
BEGIN
```

```
  P := PHI/100.0;
```

```
  CASE PLANET OF
```

```
    MERCURY : MAG := -0.42 + ( 3.80 - ( 2.73 - 2.00*P ) * P ) * P;
```

```
    VENUS   : MAG := -4.40 + ( 0.09 + ( 2.39 - 0.65*P ) * P ) * P;
```

```
    EARTH   : MAG := -3.86;
```

```
    MARS    : MAG := -1.52 + 1.6 * P;
```

```
    JUPITER : MAG := -9.40 + 0.5 * P;
```

```
    SATURN  : BEGIN
```

```
      SD := ABS ( SN(DEC) );
```

```
      DL := ABS(DLONG/100.0); IF (DL>1.6) THEN DL:=ABS(DL-3.6);
```

```
      MAG := -8.88 - 2.60 * SD + 1.25 * SD*SD + 4.40 * DL;
```

```
    END;
```

```
    URANUS  : MAG := -7.19;
```

```
    NEPTUNE : MAG := -6.67;
```

```
    PLUTO   : MAG := -1.0;
```

```
  END;
```

```
  BRIGHT := MAG + 5.0 * LN ( R*DELTA ) / LN10;
```

```
END;
```

```
(*-----*)
```

7.2.4 Apparent Diameter

The apparent diameter of a planet is the angle that the planet subtends at the Earth. It is obtained from the equatorial radius R_{Eq} of the planet at a distance Δ as

$$\varnothing_{Eq} = 2 \cdot \arcsin \left(\frac{R_{Eq}}{\Delta} \right) . \quad (7.15)$$

With planets that have polar flattening, we have to differentiate between the equatorial and polar diameters. The apparent polar diameter is always smaller than the apparent equatorial diameter and depends both on the flattening and on the planetocentric latitude $\varphi'_{\oplus}$ of the Earth. For small values of flattening, the apparent polar diameter may be obtained, to a first approximation, from

$$\varnothing_{Pol} = (1 - f \cos^2 \varphi'_{\oplus}) \varnothing_{Eq} . \quad (7.16)$$

7.3 The PHYS Program

The PHYS program calculates a table of physical ephemerides for the major planets and the Sun. It uses the subprograms previously described to determine the planetographic coordinates of the Earth, the illumination conditions and the position angle of the north pole of the rotational axis.

The coordinates of the bodies that are required are calculated by the POSITION procedure in accordance with the two-body problem, for the equinox J2000. The effects of nutation and aberration are small enough for them to be ignored in the context of the limited accuracy that is required for practical observational purposes. On the other hand, the light-time between the planet and the Earth does have to be taken into account. Consequently, the first step is to determine the geometrical distance between the Earth and the body, from which the light-time is derived. Later in the calculation, the coordinates and rotation angle that are derived are based upon the time at which the light is emitted by the body.

We do not need to take account of precession to the current equinox of date in determining the planetographic or heliographic coordinates of the Earth. These values are determined by the relative positions of the celestial bodies and are thus independent of the coordinate system used. Only in calculating the position angle, which is with respect to the actual direction of North, and thus to the equinox of date, do we need to carry out an appropriate coordinate transformation.

After the physical ephemerides for the planets have been calculated, the Sun's apparent diameter, the position angle of the axis and the heliographic coordinates of the Earth are determined. The coordinates α_0 and δ_0 of the Sun's rotation axis and the orientation of its prime meridian W are passed directly to ROTATION. The geocentric coordinates of the Sun (which are also required for the calculation of the position angle) are obtained by inverting the signs of the heliocentric coordinates of the Earth. In addition, it must be noted that heliographic longitude is measured in the opposite direction to the way in which planetographic longitudes are defined. The Earth's heliographic longitude always increases. To avoid having to incorporate a method of differentiating for this specific case, ROTATION is called with RETROGRADE as argument for the direction of rotation. In addition, it is the normal convention to give the position angle of the Sun's axis as lying within the range $-180^\circ \dots 180^\circ$, and this is taken into account by special instructions within the program.

```

(*-----*)
(*                      PHYS                      *)
(*      Physical ephemerides of the major planets and the Sun      *)
(*                      Version 93/07/01                      *)
(*-----*)

```

```
PROGRAM PHYS ( INPUT, OUTPUT );
```

```

CONST AU      = 149597870.0; (* 1 AU in km          *)
      C_LIGHT = 173.14;      (* Velocity of light ( AU/d ) *)
      J2000    = 0.0;        (* Reference epoch J2000      *)

TYPE REAL3     = ARRAY[1..3] OF REAL;
      REAL33    = ARRAY[1..3] OF REAL3;
      PLANET_TYPE = ( MERCURY, VENUS, EARTH, MARS,
                      JUPITER, SATURN, URANUS, NEPTUNE, PLUTO );
      ROTATION_TYPE = ( DIRECT, RETROGRADE );
      SYSTEM_TYPE  = ( SYS_I, SYS_II, SYS_III );

```

```

VAR  PLANET      : PLANET_TYPE;
      DAY,MONTH,YEAR : INTEGER;
      HOUR, T,TO   : REAL;
      X,Y,Z, XE,YE,ZE : REAL;
      XX,YY,ZZ, R, DELTA : REAL;
      R_EQU, F, D_EQU : REAL;
      DD, DM       : INTEGER;
      DS           : REAL;
      L1,L2,L3, B, BSUN, D : REAL;
      AO,DO,W1,W2,W3 : REAL;
      SENSE         : ROTATION_TYPE;
      AX,AY,AZ, POSAX : REAL;
      ELONG, PHI,K, MAG : REAL;
      LSUN, DSUN, POSSUN : REAL;
      PMAT          : REAL33;

```

```

(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS, TN, ACS, ATN, ATN2, DOT, DMS                               *)
(* MJD                                                                    *)
(* ECLEQU, GAUSVEC, ORBECL                                              *)
(* PMATEQU, PRECART                                                      *)
(* ECCANOM, ELLIP                                                        *)
(* POSITION                                                                *)
(* SHAPE, ORIENT, ROTATION, POSANG, BRIGHT, ILLUM                      *)
(*-----*)

```

```
BEGIN
```

```

(* Print header and read desired date *)
WRITELN;
WRITELN ( ' PHYS: physical ephemerides of the planets and the sun' );
WRITELN ( '                      Version 93/07/01                      ' );
WRITELN ( '          (c) 1993 Thomas Pflieger, Oliver Montenbruck          ' );

```

```

WRITELN;
WRITE  (' Date (yyyy mm dd hh.hhh) ');
READLN  (YEAR,MONTH,DAY,HOURL);
WRITELN;

T := ( MJD(DAY,MONTH,YEAR,HOURL)-51544.5 ) / 36525.0;

WRITE  ('D':13,'V':7,'i':7,'PA(S)':9,'PA(A)':6);
WRITELN ('L(I)':8,'L(II)':7,'L(III)':7,'B':6);
WRITELN ('":13,'mag':8,'o':6,'o':7,'o':8,'o':8,'o':7,'o':8);

(* Precession matrix (J2000 -> mean equinox of date) *)
(* for equatorial coordinates *)

PMATEQU (J2000, T, PMAT);

(* Equatorial coordinates of the Earth, mean equinox of J2000 *)

POSITION ( EARTH, T, XE,YE,ZE );
ECLEQU   ( J2000, XE,YE,ZE );

(* Physical ephemerides of the planets *)

FOR PLANET:=MERCURY TO PLUTO DO
  IF PLANET<>EARTH THEN

    BEGIN

      (* Compute the planet's geocentric geometric position and the *)
      (* light time (in days) *)

      POSITION ( PLANET, T, X,Y,Z );
      ECLEQU   ( J2000, X,Y,Z );
      DELTA := SQRT ( (X-XE)*(X-XE) + (Y-YE)*(Y-YE) + (Z-ZE)*(Z-ZE) );
      TO := T - (DELTA/C_LIGHT) / 36525.0;

      (* Compute the antedated planetary position at emission of light *)
      (* (i.e. apply a first order light time correction) *)

      POSITION ( PLANET, TO, X,Y,Z );
      ECLEQU   ( J2000, X,Y,Z );

      (* Light time corrected geocentric coordinates *)

      XX := X-XE; YY := Y-YE; ZZ := Z-ZE;

      (* Compute apparent equatorial diameter (in ") *)

      SHAPE ( PLANET, R_EQU, F );
      D_EQU := 3600.0 * 2.0*ASN(R_EQU/(DELTA*AU));

      (* Compute right ascension and declination of the axis of *)

```

```

(* rotation with respect to the equator and equinox of J2000; *)
(* compute orientation of the prime meridian *)

ORIENT ( PLANET,SYS_I ,T0, A0,D0,W1,SENSE );
ORIENT ( PLANET,SYS_II ,T0, A0,D0,W2,SENSE );
ORIENT ( PLANET,SYS_III,T0, A0,D0,W3,SENSE );

(* Compute planetocentric longitude and latitude of the Earth *)

ROTATION ( XX,YY,ZZ,A0,D0,W1,SENSE,F, AX,AY,AZ,L1,B,D );
ROTATION ( XX,YY,ZZ,A0,D0,W2,SENSE,F, AX,AY,AZ,L2,B,D );
ROTATION ( XX,YY,ZZ,A0,D0,W3,SENSE,F, AX,AY,AZ,L3,B,D );

(* Compute planetocentric longitude and latitude of the Sun *)

ROTATION ( X,Y,Z,A0,D0,W1,SENSE,F, AX,AY,AZ,LSUN,BSUN,DSUN );

(* Compute illumination and apparent magnitude *)

ILLUM ( X,Y,Z, XE,YE,ZE, R,DELTA,ELONG, PHI,K );
MAG := BRIGHT ( PLANET, R,DELTA,PHI,DSUN,LSUN-L1 );

(* Compute position angles of the axis of rotation and of the *)
(* Sun with respect to the mean equinox of date *)

PRECART ( PMAT, X, Y, Z );
PRECART ( PMAT, XX,YY,ZZ );
PRECART ( PMAT, AX,AY,AZ );

POSAX := POSANG ( XX,YY,ZZ, AX,AY,AZ );
POSSUN := POSANG ( XX,YY,ZZ, -X,-Y,-Z );

(* Print results *)

CASE PLANET OF
  MERCURY: WRITE ( ' Mercury ' );
  VENUS : WRITE ( ' Venus ' );
  MARS : WRITE ( ' Mars ' );
  JUPITER: WRITE ( ' Jupiter ' );
  SATURN : WRITE ( ' Saturn ' );
  URANUS : WRITE ( ' Uranus ' );
  NEPTUNE: WRITE ( ' Neptune ' );
  PLUTO : WRITE ( ' Pluto ' );
END;

WRITE ( D_EQU:5:2, MAG:6:1, PHI:7:1 );
WRITE ( POSSUN:8:2, POSAX:8:2 );

CASE PLANET OF
  MERCURY,VENUS,MARS,PLUTO: WRITE (L1:8:2, ' ':14);
  JUPITER: WRITE (L1:8:2, L2:7:2, L3:7:2);
  SATURN: WRITE (L1:8:2, L3:14:2);
  URANUS,NEPTUNE: WRITE (L3:22:2)

```



```

END;

WRITELN (B:8:2);

END;

(* Physical ephemerides of the Sun *)

(* Compute light time corrected equatorial coordinates of      *)
(* the Earth with respect to the equator and equinox of J2000 *)

DELTA := SQRT ( XE*XE + YE*YE + ZE*ZE );
TO := T - (DELTA/C_LIGHT) / 36525.0;
POSITION ( EARTH, TO, XE,YE,ZE );
ECLEQU   ( J2000, XE,YE,ZE );

(* Right ascension and declination of the Sun's axis (J2000),  *)
(* orientation of the prime meridian and equatorial radius (km) *)

AO := 285.96;
DO := 63.96;
W1 := 84.11 + 14.1844000 * 36525.0*T;
W1 := W1/360.0; W1:=360.0*(W1-TRUNC(W1));
R_EQU := 696000.0;

(* Compute heliographic coordinates of the Earth *)

ROTATION ( -XE,-YE,-ZE, AO,DO,W1, RETROGRADE, 0.0, AX,AY,AZ,L1,B,B );

(* Compute position angle of the axis of rotation *)
(* with respect to the mean equinox of date      *)

PRECART ( PMAT, XE,YE,ZE );
PRECART ( PMAT, AX,AY,AZ );
POSAX := POSANG ( -XE,-YE,-ZE, AX,AY,AZ );

(* Express position angle of the Sun's axis within -180..180 deg *)

IF POSAX>180.0 THEN POSAX:=POSAX-360.0;

(* Compute apparent equatorial diameter (in ") *)

D_EQU := 2.0*ASN(R_EQU/(DELTA*AU));

(* Print results *)

DMS (D_EQU, DD,DM,DS);
WRITELN ( ''':10, ''':3, 'o':29, 'o':8, 'o':22);
WRITELN ( ' Sun ', DM:3, DS:6:2, POSAX:29:2, L1:8:2, B:22:2 )

END.

(*-----*)

```

As an example we will calculate the physical ephemerides for 1993 March 30 at 0^h Ephemeris Time. As usual, the input data is shown in *italic*.

The following are displayed: the apparent equatorial diameter *D* in arc-seconds, the visual magnitude *V*, the phase angle *i*, as well as the position angle of the Sun *PA(S)* and of the north pole *PA(A)*. In addition the planetographic coordinates of the Earth are given in the rotation systems appropriate for each planet.

PHYS: physical ephemerides of the planets and the Sun

Version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Date (yyyy mm dd hh.hhh) 1993 3 30 0.0

	D	V	i	PA(S)	PA(A)	L(I)	L(II)	L(III)	B
	"	mag	°	°	°	°	°	°	°
Mercury	8.59	0.6	103.9	65.95	334.57	143.78			-5.63
Venus	58.96	-4.1	167.4	183.77	337.04	334.11			-9.29
Mars	7.85	0.5	36.5	278.01	343.26	273.00			8.62
Jupiter	44.20	-2.5	0.3	42.10	24.98	136.71	227.76	172.79	-3.18
Saturn	15.68	0.8	4.0	68.85	6.29	41.37		124.91	14.47
Uranus	3.54	5.8	2.9	80.70	275.12			195.18	-56.54
Neptune	2.21	7.9	1.9	81.35	2.32			228.40	-29.45
Pluto	0.14	13.7	1.4	89.32	84.47	289.36			-14.43
	, "				°	°			°
Sun	32	1.81			-26.00	185.12			-6.56

Venus is close to inferior conjunction, a few degrees north of the Sun, and shows the very large, narrow crescent ($i = 167.4^\circ$) phase that may sometimes be seen, even in daylight, under very favourable conditions. The direction from which the sunlight is falling is almost due south ($\vartheta_0 = 183.8^\circ$).

Jupiter is conspicuous because of its large apparent diameter and its brightness. It is at opposition, so its phase angle is extremely small ($i = 0.3^\circ$).

Mars, on the other hand, is already past opposition and exhibits a distinct phase ($i = 36.5^\circ$). In addition, its small diameter of slightly less than eight arc-seconds means that recognizing any features on its surface is very difficult.

8. The Orbit of the Moon

8.1 General Description of the Lunar Orbit

The motion of the Moon is principally determined by two bodies, the Earth and the Sun. If we consider the gravitational forces that affect the Moon, we find that it is not its nearest neighbour, the Earth, that has the greatest effect, but the more distant Sun. Although gravitational force decreases as the square of the distance, the Sun's gravity exceeds that of the Earth, because of its far greater mass. Putting figures to this (Earth-Moon distance $r \approx 380\,000$ km, Sun-Moon distance $R \approx 150$ million km, Sun/Earth mass ratio $M/m \approx 330\,000$), we find that the Sun's gravitational attraction is about twice as great as that of the Earth:

$$\frac{F_{\odot}}{F_{\oplus}} = \frac{M}{R^2} \bigg/ \frac{m}{r^2} \approx 2 \quad .$$

Regardless of the relative positions of the Sun, Earth, and Moon, the sum of the two forces therefore always has a component directed towards the Sun, and never away from it. The Moon may thus be described as following an elliptical orbit around the Sun at a distance of some 150 million km, superimposed upon which there are small monthly oscillations. Because the gravitational force is never directed away from the Sun, the orbit, despite these oscillations, is always concave towards the Sun.

Although the Moon's orbit is primarily determined by the Sun's gravity, it does not seem to be particularly advisable to describe it in heliocentric coordinates. We are, after all, primarily interested in its motion around the Earth. Because the Earth and the Moon are relatively close neighbours, they are subject to almost identical gravitational forces exerted by the Sun. If these forces were exactly equal, then the Sun would have no overall influence on the *relative* positions of the Earth and the Moon. It would only be responsible for their common yearly orbit around the Sun, while the Earth's gravity would determine the Moon's monthly motion. The difference between the forces exerted by the Sun on the Moon and on the Earth is about r/R times the amount of the force itself, and is thus about 200 times smaller than the force that the Earth exerts on the Moon. The most useful description of the Moon's orbit is therefore one expressed in a geocentric reference frame. In it, the Moon moves around the Earth in a monthly Keplerian orbit, which is perturbed to a greater or lesser degree by the Sun.

The mean orbital ellipse has an eccentricity of $e = 0.055$ and is inclined to the ecliptic at about 5.1° . The orbital period from perigee to perigee (known as the *anomalistic* month) has an average value of 27.55 days. This is almost exactly 2 days less than the time between the dates of two successive New or Full phases (the *synodic* month), 29.53 days. This difference arises from the apparent motion of the Sun against the background sky, which amounts to about 30° in one month. After one orbit around the Earth, the Moon requires roughly another two days to catch up with the Sun, and show the same phase.

Because of the eccentric orbit, the Moon varies by up to 6.3 from its mean position. This most apparent irregularity is known as the *major inequality*. It has nothing to do with solar perturbations, but is solely caused by the elliptical shape of the orbit. The amount of the major inequality follows from the equation of the centre (6.1), which was described in connection with the series expansions of planetary orbits. Ptolemy pointed out in the *Almagest* that it alone was not sufficient to explain the Moon's orbit satisfactorily. He was the first to notice *evection*, a perturbation amounting to 1.3 , which depends on the relative location of the Sun and Moon. Considerably later, two further perturbations (*variation* and *annual inequality*) were discovered by Tycho Brahe from his more accurate observations, before Newton provided the basis for a theoretical understanding of lunar motion with his theory of universal gravitation. In 1770, Mayer published the first lunar tables that were sufficiently accurate for determining position and time at sea. Nowadays a total of more than one thousand individual periodic perturbation terms are known.

As well as the greater or lesser periodic oscillations in the Moon's orbit, the influence of the Sun also causes a series of secular effects, the most important of which is a slow rotation of the plane of the lunar orbit. The line of the nodes, i.e., the intersection between the lunar orbit and the ecliptic, rotates by a full 360° in 18.6 years, while the inclination remains almost fixed. The rotation of the line of nodes is *retrograde*, which means that the longitude of the ascending node *decreases* relative to the vernal equinox by about 20° per year. The reason for this shift is that the Sun's gravity constantly tries to pull the Moon down towards the ecliptic and to force the axis of symmetry of the orbit to lie at right-angles to that plane. Like the axis of a gyroscope, however, this is not aligned with the force, so it begins a slow precessional motion, which causes it to rotate once around the pole of the ecliptic in the 18.6 years already mentioned.

The position of perigee, i.e., the orientation of the semi-major axis of the orbit in space, also varies. Perigee has a direct (or prograde) shift along the orbit of more than 40° per year. A complete rotation takes about 8.5 years.

The various parameters for the mean motion of the Sun and the Moon that have to be taken into account in theories of lunar motion, with the

customary designations and with

$$T = \frac{\text{JD} - 2451545}{36525}$$

are:

- the mean longitude of the Moon (L_0)

$$\begin{aligned} L_0 &= 218^\circ 31' 16.17'' + 481267^\circ 88' 08.8'' \cdot T - 4'' 06 \cdot T^2 \\ &= 0^\circ 60643382 + 1336^\circ 85522467 \cdot T - 0^\circ 00000313 \cdot T^2, \end{aligned} \quad (8.1)$$

- the Moon's mean anomaly (l)

$$\begin{aligned} l &= 134^\circ 96' 29.2'' + 477198^\circ 86' 75.3'' \cdot T + 33'' 25 \cdot T^2 \\ &= 0^\circ 37489701 + 1325^\circ 55240982 \cdot T + 0^\circ 00002565 \cdot T^2, \end{aligned} \quad (8.2)$$

- the Sun's mean anomaly (l')

$$\begin{aligned} l' &= 357^\circ 52' 54.3'' + 35999^\circ 04' 44.4'' \cdot T - 0'' 58 \cdot T^2 \\ &= 0^\circ 99312619 + 99^\circ 99735956 \cdot T - 0^\circ 00000044 \cdot T^2, \end{aligned} \quad (8.3)$$

- the mean distance of the Moon from the ascending node (F)

$$\begin{aligned} F &= 93^\circ 27' 28.3'' + 483202^\circ 01' 87.3'' \cdot T - 11'' 56 \cdot T^2 \\ &= 0^\circ 25909118 + 1342^\circ 22782980 \cdot T - 0^\circ 00000892 \cdot T^2, \end{aligned} \quad (8.4)$$

- and the mean elongation of the Moon (D), i.e., the difference between the mean longitudes of the Sun and the Moon,

$$\begin{aligned} D &= 297^\circ 85' 02.7'' + 445267^\circ 11' 13.5'' \cdot T - 5'' 15 \cdot T^2 \\ &= 0^\circ 82736186 + 1236^\circ 85308708 \cdot T - 0^\circ 00000397 \cdot T^2. \end{aligned} \quad (8.5)$$

The individual values are given in radians and units of one revolution ($1^\circ = 360^\circ$). L_0 refers to the mean equinox of date whereas the remaining quantities are angles, which are not affected by precession. The longitude of the ascending node Ω is not explicitly employed. It is obtained from the difference $\Omega = L_0 - F$.

The true ecliptic longitude of the Moon (λ) differs from the mean longitude by a series of periodic terms, which have l , l' , F and D as arguments:

$$\lambda = L_0 + \Delta\lambda$$

where

$$\begin{aligned}
 \Delta\lambda = & +22640'' \cdot \sin(l) && \text{(major inequality)} \\
 & -4586'' \cdot \sin(l - 2D) && \text{(evection)} \\
 & +2370'' \cdot \sin(2D) && \text{(variation)} \\
 & +769'' \cdot \sin(2l) && \text{(major inequality)} \\
 & -668'' \cdot \sin(l') && \text{(annual inequality)} \\
 & -412'' \cdot \sin(2F) && \text{(reduction to the ecliptic)} \\
 & -212'' \cdot \sin(2l - 2D) \\
 & -206'' \cdot \sin(l + l' - 2D) \\
 & +192'' \cdot \sin(l + 2D) \\
 & -165'' \cdot \sin(l' - 2D) \\
 & +148'' \cdot \sin(l - l') \\
 & -125'' \cdot \sin(D) && \text{(parallactic inequality)} \\
 & -110'' \cdot \sin(l + l') \\
 & -55'' \cdot \sin(2F - 2D) \quad .
 \end{aligned}$$

For an unperturbed orbit, the ecliptic latitude β depends on

$$\begin{aligned}
 \sin \beta &= \sin i \cdot \sin u \\
 \beta &\approx i \cdot \sin u
 \end{aligned}$$

that is, on the orbital inclination i and the position in the orbit. Here u denotes the argument of latitude, i.e. the angle between the position vector and the line of nodes. Taking perturbations by the Sun into account, the latitude of the Moon may be expressed, to a first approximation, by a slightly altered equation, namely:

$$\beta \approx 18520'' \sin(S) + N$$

where $S = F + \Delta S$. ΔS consists of a series of terms, which are mainly the same as in the expansion for $\Delta\lambda$. If we consider only the most important differences, we have:

$$\Delta S \approx \Delta\lambda + 412'' \sin(2F) + 541'' \sin(l') \quad .$$

The quantity N incorporates a small number of additional latitude variations, which are caused by an oscillation of the inclination of the orbit:

$$N = -526'' \sin(F - 2D) + 44'' \sin(l + F - 2D) - 31'' \sin(-l + F - 2D) \dots \quad (8.6)$$

This greatly simplified description will serve to give a rough idea of the way lunar theory is expressed and how the Moon's ecliptic longitude and latitude are calculated. The equations just described are evaluated in the `MINI_MOON` program, which was given in Chap. 3 without detailed discussion. A glance at this program should clarify the individual steps taken in the calculation.

A point that should be mentioned is that we have knowingly avoided using a series expansion of β itself to calculate the ecliptic latitude. Such a series, whose most important terms are

$$\beta = 18461'' \sin(F) + 1010'' \sin(l+F) + 1000'' \sin(l-F) - 624'' \sin(F-2D) \\ - 199'' \sin(l-F-2D) - 167'' \sin(l+F-2D) + \dots,$$

is certainly easier to employ, but requires, an unnecessarily high expenditure of computational effort. This is because the individual terms contain just odd multiples of F , whereas the terms in the expansions of $\Delta\lambda$ and ΔS always incorporate only even multiples of F . For each term in $\Delta\lambda$, there is a corresponding term in ΔS , requiring the same angular function. If the program is appropriately arranged, therefore, no additional trigonometric terms have to be evaluated in calculating ΔS . Conversely, the series for $\Delta\lambda$ and $\Delta\beta$ have no terms that can be evaluated together.

8.2 Brown's Lunar Theory

The lunar theory developed by E.W. Brown at the beginning of the 20th century is one of the best-known analytical expressions of lunar motion. The version presented here is the Nautical Almanac Office's *Improved Lunar Ephemeris* of 1954. Only a fraction of the theory's more than one thousand perturbation terms are used in the MOON program. The accuracy thus obtainable is about one second of arc.

The calculation begins by determining the mean arguments L , l , l' , F and D from (8.1) to (8.5). The time T is expressed as Julian centuries *Ephemeris Time* since epoch J2000. The mean arguments are also subject to small long-period oscillations, which are to be added to the values given above as corrections:

ΔL_0	Δl	$\Delta l'$	ΔF	ΔD	
+0''84	+2''94	-6''40	+0''21	+7''24	$\cdot \sin(2\pi(0.19833 + 0.05611T))$
+0''31	+0''31	0''0	+0''31	+0''31	$\cdot \sin(2\pi(0.27869 + 0.04508T))$
+14''27	+14''27	0''0	+14''27	+14''27	$\cdot \sin(2\pi(0.16827 - 0.36903T))$
+7''26	+9''34	0''0	-88''70	+7''26	$\cdot \sin(2\pi(0.34734 - 5.37261T))$
+0''28	+1''12	0''0	-15''30	+0''28	$\cdot \sin(2\pi(0.10498 - 5.37899T))$
+0''24	+0''83	-1''89	+0''24	+2''13	$\cdot \sin(2\pi(0.42681 - 0.41855T))$
0''00	0''00	0''00	-1''86	0''00	$\cdot \sin(2\pi(0.14943 - 5.37511T))$

(8.7)

Using the improved mean arguments thus obtained, five series of perturbation terms are calculated:

$\Delta\lambda$	perturbations in ecliptic longitude,
$\Delta S, N, \gamma_1 C$	perturbations in ecliptic latitude and
$\Delta \sin \Pi$	perturbations in parallax .

All these series have a common structure:

$$\left\{ \begin{matrix} \Delta\lambda, \Delta S, N \\ \gamma_1 C, \Delta \sin II \end{matrix} \right\} = \sum_n \left\{ \begin{matrix} a_n, b_n, c_n \\ d_n, e_n \end{matrix} \right\} \cdot \left\{ \begin{matrix} \sin \\ \cos \end{matrix} \right\} (p_n l + q_n l' + r_n F + s_n D) \quad .$$

The array (p, q, r, s) describes how an individual summand depends on l, l', F and D and hence the periodicity of the term. An effective way of carrying out the calculation is to combine all the terms that have similar characteristics. The following is just a small portion of the whole table of perturbation terms:

$\Delta\lambda$	ΔS	$\gamma_1 C$	$\Delta \sin II$	p	q	r	s
+22639"500	+22609"07	+0"079	+186"5398	+1	0	0	0
-4586"465	-4578"13	-0"077	+34"3117	+1	0	0	-2
+2369"912	+2373"36	+0"601	+28"2333	0	0	0	+2
+769"016	+767"96	+0"107	+10"1657	+2	0	0	0
-668"146	-126"98	-1"302	-0"3997	0	+1	0	0
-411"608	-0"20	+0"000	-0"0124	0	0	+2	0

Because of its different coefficients, the series for the quantity N cannot be calculated at the same time as the other series. However, it contains only a few terms.

Because of the large number of perturbation terms, it is worthwhile taking some pains over the calculation of the various angular functions that occur. The calculation of a sine or cosine function – when compared with elementary operations like the four basic functions – requires a relatively large amount of effort and time. It is not, however, necessary to calculate every angular function explicitly, if use is made of the addition theorems

$$\begin{aligned} \cos(\alpha_1 + \alpha_2) &= \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 \\ \sin(\alpha_1 + \alpha_2) &= \sin \alpha_1 \cos \alpha_2 + \cos \alpha_1 \sin \alpha_2 \quad . \end{aligned}$$

Using the short ADDTHE routine

```
PROCEDURE ADDTHE(C1,S1,C2,S2:REAL;VAR C,S:REAL);
  BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;
```

we can first calculate sines and cosines for multiples of the mean arguments:

$$CO[I,K] = \begin{cases} \cos(il) & (k=1) \\ \cos(il') & (k=2) \\ \cos(iF) & (k=3) \\ \cos(iD) & (k=4) \end{cases} \quad SI[I,K] = \begin{cases} \sin(il) & (k=1) \\ \sin(il') & (k=2) \\ \sin(iF) & (k=3) \\ \sin(iD) & (k=4) \end{cases} \quad .$$

For l ($K=1$) and using

```
CO[0,K]:=1.0; CO[1,K]:=COS(L); SI[0,K]:=0.0; SI[1,K]:=SIN(L);
FOR I := 2 TO MAX DO
  ADDTHE(CO[I-1,K],SI[I-1,K],CO[1,K],SI[1,K],CO[I,K],SI[I,K]);
FOR I := 1 TO MAX DO
  BEGIN CO[-I,K]:=CO[I,K]; SI[-I,K]:=-SI[I,K]; END;
```


we can obtain all the values of $\cos(il)$ and $\sin(il)$ for $I=-MAX...+MAX$ with only two calls of the sine or cosine functions. The procedure **TERM**, which uses the values of $CO[I,K]$ and $SI[I,K]$, then allows

$$x = \cos(pl + ql' + rF + sD) \quad \text{and} \quad y = \sin(pl + ql' + rF + sD)$$

to be calculated simply for given coefficients (p, q, r, s) .

```

PROCEDURE TERM(P,Q,R,S:INTEGER;VAR X,Y:REAL);
  VAR I: ARRAY[1..4] OF INTEGER; K: INTEGER;
  BEGIN
    I[1]:=P; I[2]:=Q; I[3]:=R; I[4]:=S; X:=1.0; Y:=0.0;
    FOR K:=1 TO 4 DO
      IF (I[K]>0) THEN ADDTHE(X,Y,CO[I[K],K],SI[I[K],K],X,Y);
    END;
  
```

The application of this procedure can be explained very clearly by taking the series for the latitude perturbation N as an example. The procedure **SOLARN** is designed to sum all the terms in the perturbation series mentioned into the variable N . An individual term is evaluated in the sub-routine **ADDN**. The parameters passed to this sub-routine are the coefficient and the characteristic values (p, q, r, s) of the term. Using **TERM**, the value of the associated angular functions are calculated. The sine terms is then multiplied by the coefficients and added to N . Naturally, before **SOLARN** is called, all the values of the arrays $CO[I,K]$ and $SI[I,K]$ must be determined.

```

PROCEDURE SOLARN(VAR N: REAL);
  VAR X,Y: REAL;
  PROCEDURE ADDN(COEFFN:REAL;P,Q,R,S:INTEGER);
    BEGIN TERM(P,Q,R,S,X,Y); N:=N+COEFFN*Y END;
  BEGIN
    N := 0.0;
    ADDN(-526.069, 0, 0,1,-2); ADDN( -3.352, 0, 0,1,-4);
    ADDN(+44.297,+1, 0,1,-2); ADDN( -6.000,+1, 0,1,-4);
    ADDN(+20.599,-1, 0,1, 0); ADDN( -30.598,-1, 0,1,-2);
    ADDN( -24.649,-2, 0,1, 0); ADDN( -2.000,-2, 0,1,-2);
    ADDN( -22.571, 0,+1,1,-2); ADDN( +10.985, 0,-1,1,-2);
  END;
BEGIN
  ... CO and SI are calculated here ...
  SOLARN(N); WRITELN(N);
  ...
END;
  
```

The **ADD SOL**, **SOLAR1**, **SOLAR2**, and **SOLAR3** procedures, which calculate the series $\Delta\lambda$, ΔS , $\gamma_1 C$, and $\Delta \sin H$, function in precisely the same way.

The treatment of the solar perturbation terms in the method just described is not completely comprehensive, however. The reason for this is that, strictly speaking, the coefficients of the perturbation terms are slightly different from the values quoted. Each coefficient in the series for $\Delta\lambda$, ΔS , N ,

$\gamma_1 C$, and $\Delta \sin \Pi$ is actually a function of various parameters—such as the eccentricity of the solar orbit, for example—for which Brown took specific numerical values. In order to take the correct values for these parameters into account, each term (which is characterized by the array (p, q, r, s)) must be multiplied by a factor

$$(1.000002208)^{|p|} \cdot (1.0 - 0.002495388(T+1))^{|q|} \cdot (1.000002708 + 139.978\Delta\gamma)^{|r|}.$$

Here $\Delta\gamma$ is given by the value of

$$\begin{aligned}\Delta\gamma = & -0.000003332 \cdot \sin(2\pi(0.59734 - 5.37261T)) \\ & -0.000000539 \cdot \sin(2\pi(0.35498 - 5.37899T)) \\ & -0.000000064 \cdot \sin(2\pi(0.39943 - 5.37511T))\end{aligned}$$

which is calculated in LONG_PERIODIC. Anyway, these corrections can be taken into account very simply, if in calculating the arrays CO[I,K] and SI[I,K] as described above, we multiply the values

$$\begin{array}{lll}\text{CO}[1,1] \text{ and SI}[1,1] & \text{by} & (1.000002208), \\ \text{CO}[1,2] \text{ and SI}[1,2] & \text{by} & (1.0 - 0.002495388(T+1)) \text{ and} \\ \text{CO}[1,3] \text{ and SI}[1,3] & \text{by} & (1.000002708 + 139.978\Delta\gamma) \quad .\end{array}$$

The appropriate correction factors will already have been taken into account in all the previously calculated angular functions, therefore, when the repeated calls to ADDTHE are made. For example, SI[3,2] contains the value of $(1.0 - 0.002495388(T+1))^{|3|} \cdot \sin(3I')$ instead of $\sin(3I')$. This small change enables all the terms to be automatically, and without great difficulty, multiplied by the right factors.

This concludes the calculation of the various lunar perturbations that are caused by the Sun. However, the Sun is not the only body in the Solar System that affects the orbit of the Moon. A complete theory of the lunar orbit cannot neglect planetary perturbations. In the MOON program the most important contributions by Venus and Jupiter are also taken into account:

$$\begin{aligned}\Delta\lambda_{\text{plan}} = & +0''.82 \sin(0^\circ 7736 - 62^\circ 5512T) + 0''.31 \sin(0^\circ 0466 - 125^\circ 1025T) \\ & + 0''.35 \sin(0^\circ 5785 - 25^\circ 1042T) + 0''.66 \sin(0^\circ 4591 + 1335^\circ 8075T) \\ & + 0''.64 \sin(0^\circ 3130 - 91^\circ 5680T) + 1''.14 \sin(0^\circ 1480 + 1331^\circ 2898T) \\ & + 0''.21 \sin(0^\circ 5918 + 1056^\circ 5859T) + 0''.44 \sin(0^\circ 5784 + 1322^\circ 8595T) \\ & + 0''.24 \sin(0^\circ 2275 - 5^\circ 7374T) + 0''.28 \sin(0^\circ 2965 + 2^\circ 6929T) \\ & + 0''.33 \sin(0^\circ 3132 + 6^\circ 3368T) \quad .\end{aligned}$$

These perturbations in ecliptic longitude are calculated in PLANETARY.

Now all the necessary values are known. The ecliptic longitude of the Moon is obtained by adding the solar and planetary perturbations to the mean longitude:

$$\lambda = L_0 + \Delta\lambda + \Delta\lambda_{\text{plan}} \quad .$$

L_0 here is the value of the mean longitude already corrected as given in (8.7). To obtain the ecliptic latitude β , ΔS is first added to the mean distance F from the node:

$$S = F + \Delta S .$$

We then have

$$\beta = (1.000002708 + 139.978\Delta\gamma) \cdot \{18519''70 + \gamma_1 C\} \cdot \sin(S) - \{6''24\} \cdot \sin(3S) + N .$$

The distance of the Moon is calculated from the sine of the horizontal parallax:

$$r = (1/\sin \Pi) R_{\oplus} \quad (\text{radius of the Earth } R_{\oplus} \approx 6378.14 \text{ km}) .$$

$\sin \Pi$ has the value

$$\sin \Pi = 0.999953253 \cdot (3422''7 + \Delta \sin \Pi) \cdot \frac{\pi}{180 \cdot 3600''} .$$

The factor

$$\frac{\pi}{180 \cdot 3600''} = \frac{1}{206264''81}$$

here merely serves to convert seconds of arc into radians.

The complete MOON procedure follows. For a given time T , (expressed in Julian centuries Ephemeris Time since epoch J2000), the longitude and latitude of the Moon are calculated in geocentric ecliptic coordinates, as well as its distance from the centre of the Earth in Earth radii. The coordinates are referred to the mean equinox of date.

```
(*-----*)
(* MOON: analytical lunar theory by E.W.Brown (Improved Lunar Ephemeris) *)
(*      with an accuracy of approx. 1"                                     *)
(*                                                                                   *)
(*      T:      time in Julian centuries since J2000 (Ephemeris Time)          *)
(*      (T=(JD-2451545.0)/36525.0)                                             *)
(*      LAMBDA: geocentric ecliptic longitude (equinox of date)                *)
(*      BETA:   geocentric ecliptic latitude  (equinox of date)                *)
(*      R:      geocentric distance (in Earth radii)                          *)
(*-----*)
```

```
PROCEDURE MOON ( T:REAL; VAR LAMBDA,BETA,R: REAL );
```

```
CONST PI2 = 6.283185308; (* 2*pi; pi=3.141592654... *)
      ARC = 206264.81;   (* 3600*180/pi = arcsec per radian *)
```

```
VAR DGAM,FAC      : REAL;
    DLAM,N,GAM1C,SINPI : REAL;
    LO, L, LS, F, D ,S : REAL;
    DLO,DL,DLS,DF,DD,DS: REAL;
    CO,SI: ARRAY[-6..6,1..4] OF REAL;
```

```

(* fractional part of a number; with several compilers it may be *)
(* necessary to replace TRUNC by LONG_TRUNC or INT if T<-24! *)
FUNCTION FRAC(X:REAL):REAL;
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;

(* calculate c=cos(a1+a2) and s=sin(a1+a2) from the addition theo- *)
(* rems for c1=cos(a1), s1=sin(a1), c2=cos(a2) and s2=sin(a2) *)
PROCEDURE ADDTHE(C1,S1,C2,S2:REAL;VAR C,S:REAL);
  BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;

(* calculate sin(phi); phi in units of 1 revolution = 360 degrees *)
FUNCTION SINE (PHI:REAL):REAL;
  BEGIN SINE:=SIN(PI2*FRAC(PHI)); END;

(* calculate long-periodic changes of the mean elements *)
(* l,l',F,D and L0 as well as dgamma *)
PROCEDURE LONG_PERIODIC ( T: REAL; VAR DLO,DL,DLS,DF,DD,DGAM: REAL );
  VAR S1,S2,S3,S4,S5,S6,S7: REAL;
  BEGIN
    S1:=SINE(0.19833+0.05611*T); S2:=SINE(0.27869+0.04508*T);
    S3:=SINE(0.16827-0.36903*T); S4:=SINE(0.34734-5.37261*T);
    S5:=SINE(0.10498-5.37899*T); S6:=SINE(0.42681-0.41855*T);
    S7:=SINE(0.14943-5.37511*T);
    DLO:= 0.84*S1+0.31*S2+14.27*S3+ 7.26*S4+ 0.28*S5+0.24*S6;
    DL := 2.94*S1+0.31*S2+14.27*S3+ 9.34*S4+ 1.12*S5+0.83*S6;
    DLS:=-6.40*S1 -1.89*S6;
    DF := 0.21*S1+0.31*S2+14.27*S3-88.70*S4-15.30*S5+0.24*S6-1.86*S7;
    DD := DLO-DLS;
    DGAM := -3332E-9 * SINE(0.59734-5.37261*T)
            -539E-9 * SINE(0.35498-5.37899*T)
            -64E-9 * SINE(0.39943-5.37511*T);
  END;

(* INIT: calculates the mean elements and their sine and cosine *)
(* l mean anomaly of the Moon l' mean anomaly of the Sun *)
(* F mean distance from the node D mean elongation from the Sun *)
PROCEDURE INIT;
  VAR I,J,MAX : INTEGER;
  T2,ARG,FAC: REAL;
  BEGIN
    T2:=T*T;
    DLAM:=0; DS:=0; GAM1C:=0; SINPI:=3422.7000;
    LONG_PERIODIC ( T, DLO,DL,DLS,DF,DD,DGAM );
    L0 := PI2*FRAC(0.60643382+1336.85522467*T-0.00000313*T2) + DLO/ARC;
    L := PI2*FRAC(0.37489701+1325.55240982*T+0.00002565*T2) + DL /ARC;
    LS := PI2*FRAC(0.99312619+ 99.99735956*T-0.00000044*T2) + DLS/ARC;
    F := PI2*FRAC(0.25909118+1342.22782960*T-0.00000892*T2) + DF /ARC;
    D := PI2*FRAC(0.82736166+1236.85308708*T-0.00000397*T2) + DD /ARC;
    FOR I := 1 TO 4 DO
      BEGIN
        CASE I OF

```

```

1: BEGIN ARG:=L; MAX:=4; FAC:=1.000002208; END;
2: BEGIN ARG:=LS; MAX:=3; FAC:=0.997504612-0.002495388*T; END;
3: BEGIN ARG:=F; MAX:=4; FAC:=1.000002708+139.978*DGAM; END;
4: BEGIN ARG:=D; MAX:=6; FAC:=1.0; END;
END;
CO[0,I]:=1.0; CO[1,I]:=COS(ARG)*FAC;
SI[0,I]:=0.0; SI[1,I]:=SIN(ARG)*FAC;
FOR J := 2 TO MAX DO
  ADDTHE(CO[J-1,I],SI[J-1,I],CO[1,I],SI[1,I],CO[J,I],SI[J,I]);
FOR J := 1 TO MAX DO
  BEGIN CO[-J,I]:=CO[J,I]; SI[-J,I]:=-SI[J,I]; END;
END;
END;

(* TERM calculates X=cos(p*arg1+q*arg2+r*arg3+s*arg4) and *)
(* Y=sin(p*arg1+q*arg2+r*arg3+s*arg4) *)
PROCEDURE TERM(P,Q,R,S:INTEGER;VAR X,Y:REAL);
  VAR I: ARRAY[1..4] OF INTEGER; K: INTEGER;
  BEGIN
    I[1]:=P; I[2]:=Q; I[3]:=R; I[4]:=S; X:=1.0; Y:=0.0;
    FOR K:=1 TO 4 DO
      IF (I[K]>0) THEN ADDTHE(X,Y,CO[I[K],K],SI[I[K],K],X,Y);
    END;
  END;

PROCEDURE ADDSOL(COEFFL,COEFFS,COEFFG,COEFFP:REAL;P,Q,R,S:INTEGER);
  VAR X,Y: REAL;
  BEGIN
    TERM(P,Q,R,S,X,Y);
    DLAM :=DLAM +COEFFL*Y; DS :=DS +COEFFS*Y;
    GAM1C:=GAM1C+COEFFG*X; SINPI:=SINPI+COEFFP*X;
  END;

PROCEDURE SOLAR1;
  BEGIN
    ADDSOL( 13.902, 14.06,-0.001, 0.2607,0, 0, 0, 4);
    ADDSOL( 0.403, -4.01,+0.394, 0.0023,0, 0, 0, 3);
    ADDSOL( 2369.912, 2373.36,+0.601, 26.2333,0, 0, 0, 2);
    ADDSOL( -125.154, -112.79,-0.725, -0.9781,0, 0, 0, 1);
    ADDSOL( 1.979, 6.98,-0.445, 0.0433,1, 0, 0, 4);
    ADDSOL( 191.953, 192.72,+0.029, 3.0861,1, 0, 0, 2);
    ADDSOL( -8.466, -13.51,+0.455, -0.1093,1, 0, 0, 1);
    ADDSOL(22639.500,22609.07,+0.079, 186.5398,1, 0, 0, 0);
    ADDSOL( 18.609, 3.59,-0.094, 0.0118,1, 0, 0,-1);
    ADDSOL(-4586.465,-4578.13,-0.077, 34.3117,1, 0, 0,-2);
    ADDSOL( +3.215, 5.44,+0.192, -0.0386,1, 0, 0,-3);
    ADDSOL( -38.428, -38.64,+0.001, 0.6008,1, 0, 0,-4);
    ADDSOL( -0.393, -1.43,-0.092, 0.0086,1, 0, 0,-6);
    ADDSOL( -0.289, -1.59,+0.123, -0.0053,0, 1, 0, 4);
    ADDSOL( -24.420, -25.10,+0.040, -0.3000,0, 1, 0, 2);
    ADDSOL( 18.023, 17.93,+0.007, 0.1494,0, 1, 0, 1);
    ADDSOL( -668.146, -126.98,-1.302, -0.3997,0, 1, 0, 0);
    ADDSOL( 0.560, 0.32,-0.001, -0.0037,0, 1, 0,-1);
    ADDSOL( -165.145, -165.06,+0.054, 1.9178,0, 1, 0,-2);
  END;

```

```

ADDSOL(  -1.877,   -6.46,-0.416,   0.0339,0, 1, 0,-4);
ADDSOL(   0.213,    1.02,-0.074,   0.0054,2, 0, 0, 4);
ADDSOL(  14.387,   14.78,-0.017,   0.2833,2, 0, 0, 2);
ADDSOL(  -0.586,   -1.20,+0.054,  -0.0100,2, 0, 0, 1);
ADDSOL( 769.016,  767.96,+0.107,  10.1657,2, 0, 0, 0);
ADDSOL(  +1.750,    2.01,-0.018,   0.0155,2, 0, 0,-1);
ADDSOL( -211.656, -152.53,+5.679,  -0.3039,2, 0, 0,-2);
ADDSOL(  +1.225,    0.91,-0.030,  -0.0088,2, 0, 0,-3);
ADDSOL( -30.773,  -34.07,-0.308,   0.3722,2, 0, 0,-4);
ADDSOL(  -0.570,   -1.40,-0.074,   0.0109,2, 0, 0,-6);
ADDSOL(  -2.921,  -11.75,+0.787,  -0.0484,1, 1, 0, 2);
ADDSOL(  +1.267,    1.52,-0.022,   0.0164,1, 1, 0, 1);
ADDSOL( -109.673, -115.18,+0.461,  -0.9490,1, 1, 0, 0);
ADDSOL( -205.962, -182.36,+2.056,  +1.4437,1, 1, 0,-2);
ADDSOL(   0.233,    0.36, 0.012,  -0.0025,1, 1, 0,-3);
ADDSOL(  -4.391,   -9.66,-0.471,   0.0673,1, 1, 0,-4);

```

END;

PROCEDURE SOLAR2;

BEGIN

```

ADDSOL(   0.283,    1.53,-0.111,  +0.0060,1,-1, 0,+4);
ADDSOL(  14.577,   31.70,-1.540,  +0.2302,1,-1, 0, 2);
ADDSOL( 147.687,  136.76,+0.679,  +1.1528,1,-1, 0, 0);
ADDSOL(  -1.089,    0.55,+0.021,    0.0 ,1,-1, 0,-1);
ADDSOL(  28.475,   23.59,-0.443,  -0.2257,1,-1, 0,-2);
ADDSOL(  -0.276,   -0.38,-0.006,  -0.0036,1,-1, 0,-3);
ADDSOL(   0.636,    2.27,+0.146,  -0.0102,1,-1, 0,-4);
ADDSOL(  -0.189,   -1.68,+0.131,  -0.0028,0, 2, 0, 2);
ADDSOL(  -7.486,   -0.66,-0.037,  -0.0086,0, 2, 0, 0);
ADDSOL(  -8.096,  -16.35,-0.740,   0.0918,0, 2, 0,-2);
ADDSOL(  -5.741,   -0.04, 0.0 ,  -0.0009,0, 0, 2, 2);
ADDSOL(   0.255,    0.0 , 0.0 ,    0.0 ,0, 0, 2, 1);
ADDSOL( -411.608,  -0.20, 0.0 ,  -0.0124,0, 0, 2, 0);
ADDSOL(   0.584,    0.84, 0.0 ,  +0.0071,0, 0, 2,-1);
ADDSOL( -55.173,  -52.14, 0.0 ,  -0.1052,0, 0, 2,-2);
ADDSOL(   0.254,    0.25, 0.0 ,  -0.0017,0, 0, 2,-3);
ADDSOL(  +0.025,   -1.67, 0.0 ,  +0.0031,0, 0, 2,-4);
ADDSOL(   1.060,    2.96,-0.166,   0.0243,3, 0, 0,+2);
ADDSOL(  36.124,   50.64,-1.300,   0.6215,3, 0, 0, 0);
ADDSOL( -13.193,  -16.40,+0.258,  -0.1187,3, 0, 0,-2);
ADDSOL(  -1.187,   -0.74,+0.042,   0.0074,3, 0, 0,-4);
ADDSOL(  -0.293,   -0.31,-0.002,   0.0046,3, 0, 0,-6);
ADDSOL(  -0.290,   -1.45,+0.116,  -0.0051,2, 1, 0, 2);
ADDSOL(  -7.649,  -10.56,+0.259,  -0.1038,2, 1, 0, 0);
ADDSOL(  -8.627,   -7.59,+0.076,  -0.0192,2, 1, 0,-2);
ADDSOL(  -2.740,   -2.54,+0.022,   0.0324,2, 1, 0,-4);
ADDSOL(   1.181,    3.32,-0.212,   0.0213,2,-1, 0,+2);
ADDSOL(   9.703,   11.67,-0.151,   0.1268,2,-1, 0, 0);
ADDSOL(  -0.352,   -0.37,+0.001,  -0.0028,2,-1, 0,-1);
ADDSOL(  -2.494,   -1.17,-0.003,  -0.0017,2,-1, 0,-2);
ADDSOL(   0.360,    0.20,-0.012,  -0.0043,2,-1, 0,-4);
ADDSOL(  -1.167,   -1.25,+0.008,  -0.0106,1, 2, 0, 0);
ADDSOL(  -7.412,   -6.12,+0.117,   0.0464,1, 2, 0,-2);

```

```

ADDSOL( -0.311, -0.65,-0.032, 0.0044,1, 2, 0,-4);
ADDSOL( +0.757, 1.82,-0.105, 0.0112,1,-2, 0, 2);
ADDSOL( +2.580, 2.32,+0.027, 0.0196,1,-2, 0, 0);
ADDSOL( +2.533, 2.40,-0.014, -0.0212,1,-2, 0,-2);
ADDSOL( -0.344, -0.57,-0.025, +0.0036,0, 3, 0,-2);
ADDSOL( -0.992, -0.02, 0.0 , 0.0 ,1, 0, 2, 2);
ADDSOL( -45.099, -0.02, 0.0 , -0.0010,1, 0, 2, 0);
ADDSOL( -0.179, -9.52, 0.0 , -0.0833,1, 0, 2,-2);
ADDSOL( -0.301, -0.33, 0.0 , 0.0014,1, 0, 2,-4);
ADDSOL( -6.382, -3.37, 0.0 , -0.0481,1, 0,-2, 2);
ADDSOL( 39.528, 85.13, 0.0 , -0.7136,1, 0,-2, 0);
ADDSOL( 9.366, 0.71, 0.0 , -0.0112,1, 0,-2,-2);
ADDSOL( 0.202, 0.02, 0.0 , 0.0 ,1, 0,-2,-4);
END;

```

PROCEDURE SOLAR3;

BEGIN

```

ADDSOL( 0.415, 0.10, 0.0 , 0.0013,0, 1, 2, 0);
ADDSOL( -2.152, -2.26, 0.0 , -0.0066,0, 1, 2,-2);
ADDSOL( -1.440, -1.30, 0.0 , +0.0014,0, 1,-2, 2);
ADDSOL( 0.384, -0.04, 0.0 , 0.0 ,0, 1,-2,-2);
ADDSOL( +1.938, +3.60,-0.145, +0.0401,4, 0, 0, 0);
ADDSOL( -0.952, -1.58,+0.052, -0.0130,4, 0, 0,-2);
ADDSOL( -0.551, -0.94,+0.032, -0.0097,3, 1, 0, 0);
ADDSOL( -0.482, -0.57,+0.005, -0.0045,3, 1, 0,-2);
ADDSOL( 0.681, 0.96,-0.026, 0.0115,3,-1, 0, 0);
ADDSOL( -0.297, -0.27, 0.002, -0.0009,2, 2, 0,-2);
ADDSOL( 0.254, +0.21,-0.003, 0.0 ,2,-2, 0,-2);
ADDSOL( -0.250, -0.22, 0.004, 0.0014,1, 3, 0,-2);
ADDSOL( -3.996, 0.0 , 0.0 , +0.0004,2, 0, 2, 0);
ADDSOL( 0.557, -0.75, 0.0 , -0.0090,2, 0, 2,-2);
ADDSOL( -0.459, -0.38, 0.0 , -0.0053,2, 0,-2, 2);
ADDSOL( -1.298, 0.74, 0.0 , +0.0004,2, 0,-2, 0);
ADDSOL( 0.538, 1.14, 0.0 , -0.0141,2, 0,-2,-2);
ADDSOL( 0.263, 0.02, 0.0 , 0.0 ,1, 1, 2, 0);
ADDSOL( 0.426, +0.07, 0.0 , -0.0006,1, 1,-2,-2);
ADDSOL( -0.304, +0.03, 0.0 , +0.0003,1,-1, 2, 0);
ADDSOL( -0.372, -0.19, 0.0 , -0.0027,1,-1,-2, 2);
ADDSOL( +0.418, 0.0 , 0.0 , 0.0 ,0, 0, 4, 0);
ADDSOL( -0.330, -0.04, 0.0 , 0.0 ,3, 0, 2, 0);

```

END;

(* part N of the perturbations of ecliptic latitude

*)

PROCEDURE SOLARN(VAR N: REAL);

VAR X,Y: REAL;

PROCEDURE ADDN(COEFFN:REAL;P,Q,R,S:INTEGER);

BEGIN TERM(P,Q,R,S,X,Y); N:=N+COEFFN*Y END;

BEGIN

N := 0.0;

```

ADDN(-526.069, 0, 0,1,-2); ADDN( -3.352, 0, 0,1,-4);
ADDN( +44.297,+1, 0,1,-2); ADDN( -6.000,+1, 0,1,-4);
ADDN( +20.599,-1, 0,1, 0); ADDN( -30.598,-1, 0,1,-2);
ADDN( -24.649,-2, 0,1, 0); ADDN( -2.000,-2, 0,1,-2);

```

```
ADDN( -22.571, 0,+1,1,-2); ADDN( +10.985, 0,-1,1,-2);
END;
```

```
(* perturbations of ecliptic latitude by Venus and Jnpiter *)
```

```
PROCEDURE PLANETARY(VAR DLAM:REAL);
```

```
BEGIN
```

```
DLAM := DLAM
```

```
+0.62*SINE(0.7736 -62.5512*T)+0.31*SINE(0.0466 -125.1025*T)
+0.35*SINE(0.5785 -25.1042*T)+0.66*SINE(0.4591+1335.8075*T)
+0.64*SINE(0.3130 -91.5680*T)+1.14*SINE(0.1480+1331.2898*T)
+0.21*SINE(0.5918+1056.5859*T)+0.44*SINE(0.5784+1322.8595*T)
+0.24*SINE(0.2275 -5.7374*T)+0.28*SINE(0.2965 +2.6929*T)
+0.33*SINE(0.3132 +6.3368*T);
```

```
END;
```

```
BEGIN
```

```
INIT; SOLAR1; SOLAR2; SOLAR3; SOLARN(N); PLANETARY(DLAM);
```

```
LAMBDA := 360.0*FRAC( (LO+DLAM/ARC) / PI2 );
```

```
S := F + DS/ARC;
```

```
FAC := 1.000002708+139.978*DGAM;
```

```
BETA := (FAC*(18518.511+1.189+GAM1C)*SIN(S)-6.24*SIN(3*S)+N)/3600.0;
```

```
SINPI := SINPI * 0.999953253;
```

```
R := ARC / SINPI;
```

```
END;
```

```
(*-----*)
```

The MOON procedure can be supplemented by MOONEQU, which calculates the true equatorial coordinates Right Ascension and Declination, instead of mean ecliptic coordinates.

```
(*-----*)
```

```
(* MOONEQU: geocentric equatorial coordinates of the Moon *)
```

```
(* referred to the true equinox of date *)
```

```
(* T time in Julian centuries ephemeris time since J2000 *)
```

```
(* ( T = (JD-2451545.0)/36525 ) *)
```

```
(* RA right ascension (deg) *)
```

```
(* DEC declination (deg) *)
```

```
(* R distance (in earth radii) *)
```

```
(*-----*)
```

```
PROCEDURE MOONEQU(T:REAL;VAR RA,DEC,R:REAL);
```

```
VAR L,B,X,Y,Z: REAL;
```

```
BEGIN
```

```
MOON(T,L,B,R); (* ecliptic coordinates (mean equinox *)
```

```
CART(R,B,L,X,Y,Z); (* of date) *)
```

```
ECLEQU(T,X,Y,Z); (* transform into equatorial coordinates *)
```

```
NOTEQU(T,X,Y,Z); (* nutation *)
```

```
POLAR(X,Y,Z,R,DEC,RA);
```

```
END;
```

```
(*-----*)
```


8.3 The Chebyshev Approximation

Despite all the tricks used in evaluating the perturbation series, the calculation of lunar coordinates using the MOON procedure is still relatively costly in time and effort. This is particularly noticeable when numerous positions of the Moon are required at any one time. In the next chapter, for example, we want to predict occultations of stars by the Moon and we shall use an iterative procedure to determine the time of conjunction of the Moon and a specific star. In order to be able to solve such computationally intensive calculations efficiently, we will now develop two procedures by which the coordinates of a body can be approximated in a form that is suitable for practical use. This will then enable us to save computing time, when more coordinate values are required, than we expend in obtaining the approximation.

One simple example of an approximation has already been discussed in Chap. 3, when we represented the altitude of the Sun or the Moon by a parabola. There, for every three points on the function, we determined a second-order polynomial, which approximately represented the change in the actual function over that interval. Similarly, we can take n different points (x_i, y_i) where $i = 1 \dots n$, and unambiguously determine a polynomial $P_n(x)$ of order $n - 1$, that fits the given points exactly. There is a whole series of procedures that can be used to ascertain what this polynomial is. Here we will only mention Lagrange's and Newton's methods.

Unfortunately, in many practical cases a polynomial determined in such a way proves to be unsuitable. The given values of the function are always correctly represented, but between these fixed points the polynomial shows variations, which are not related to the actual curve of the function that is being approximated, and these variations grow in an uncontrolled manner at each end of the interval covered. This effect is greater, the higher the order of the polynomial chosen. Usable results are therefore generally obtained with Lagrange's interpolation method only for low orders (up to about $n = 5$). We require a procedure that guarantees a smooth approximation even with high-order polynomials. We shall now show how such an approximation may be obtained. The key to this are the so-called *Chebyshev polynomials* (Fig. 8.1).

The n -th order Chebyshev polynomial (for $|x| \leq 1$) is defined as

$$T_n(x) = \cos(n \cdot \arccos x) \quad . \quad (8.8)$$

It is not immediately obvious from this trigonometric representation that we are actually dealing with a polynomial. This is seen easily only for $n = 0$ ($T_0 = 1$) and $n = 1$ ($T_1 = x$). By using the addition theorem for the cosine function, however, we can rapidly derive a recursive relation for T_n , from which it is quite obvious that (8.8) actually defines polynomials. In general,

$$\cos(\alpha + \beta) + \cos(\alpha - \beta) = 2 \cos(\alpha) \cos(\beta)$$

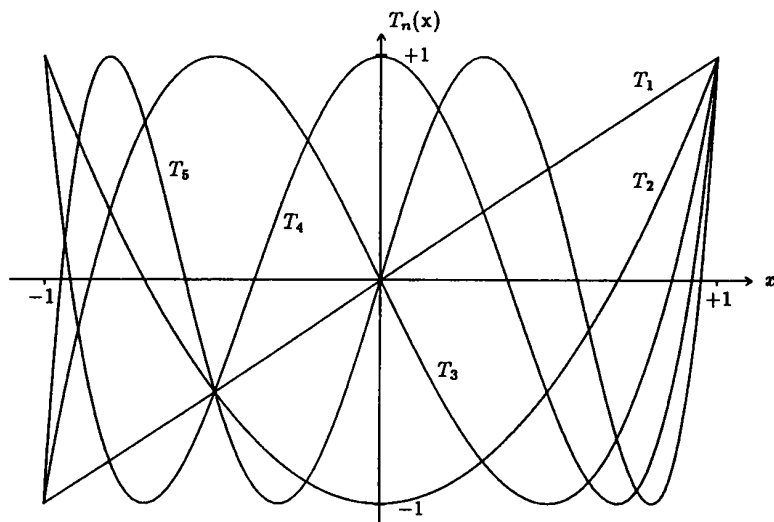


Fig. 8.1. The Chebyshev polynomials T_1 to T_6

is valid, as is

$$\cos((n+1)\varphi) = 2\cos(n\varphi)\cos(\varphi) - \cos((n-1)\varphi) \quad (8.9)$$

If we substitute $\varphi = \arccos x$, then we have

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad \text{for } n \geq 1 \quad (8.10)$$

We shall later make detailed use of this recursive expression. In any case, this relation means that $T_n(x)$ is definitely an n -th order polynomial of x . Written in full, the first Chebyshev polynomials are:

$$\begin{aligned} T_0(x) &= 1 & T_3(x) &= 4x^3 - 3x \\ T_1(x) &= x & T_4(x) &= 8x^4 - 8x^2 + 1 \\ T_2(x) &= 2x^2 - 1 & T_5(x) &= 16x^5 - 20x^3 + 5x \end{aligned}$$

We now need to look at the properties of Chebyshev polynomials in the interval $[-1, +1]$ in more detail. As can be seen easily by substitution in the definition (8.8), $T_n(x)$ becomes zero at exactly n points within this interval:

$$T_n(x) = 0 \quad \text{for } x = \cos(\pi \cdot (k-1/2)/n) \quad \text{where } k = 1, \dots, n \quad (8.11)$$

For $|x| \leq 1$, the values of $|T_n(x)|$ cannot exceed 1.

In order to approximate an arbitrary function $f(x)$ in the interval $[a, b]$, we now replace the independent variable x by the normalized variable $\hat{x}$ for the interval $[-1, +1]$. For this we require the conversions

$$\hat{x} = \frac{x - \frac{1}{2}(a+b)}{\frac{1}{2}(b-a)} \quad \text{for } x \in [a, b] \rightarrow \hat{x} \in [-1, +1]$$

and

$$x = \hat{x} \cdot \frac{1}{2}(b-a) + \frac{1}{2}(a+b) \quad \text{for } \hat{x} \in [-1, +1] \rightarrow x \in [a, b] \quad .$$

A function $f(x)$ can then be expressed in the form

$$f(x) \approx f^*(x) = \sum_{j=0}^n c_j T_j(\hat{x}) - c_0/2 \quad (8.12)$$

by Chebyshev polynomials up to order n . The coefficients c_j in this sum are calculated from

$$c_j = \frac{2}{n+1} \sum_{k=1}^{n+1} f(x_k^{n+1}) T_j(\hat{x}_k^{n+1}) \quad , \quad (8.13)$$

where $\hat{x}_k^{n+1}$ represents the k -th zero point of T_{n+1} . Written in full, the individual terms in this equation, because of (8.8) and (8.11), are:

$$\begin{aligned} x_k^{n+1} &= \frac{(b-a)}{2} \cdot \cos\left(\pi \frac{2k-1}{2n+2}\right) + \frac{(a+b)}{2} \\ T_j(\hat{x}_k^{n+1}) &= \cos\left(j\pi \frac{2k-1}{2n+2}\right) \quad . \end{aligned}$$

We can see that in order to calculate the coefficient c_j , the given function f must be evaluated at $(n+1)$ points. These are fixed as the points at which T_{n+1} is zero, and cannot be chosen freely as in the Lagrangian interpolation. This is the secret of the Chebyshev approximation's success. f^* is not just an n -th order polynomial that agrees with f at $(n+1)$ points ($f^*(x_k^{n+1}) = f(x_k^{n+1})$). By choosing points that are closer spaced at the ends of the interpolation interval than they are in the centre, a far smoother, overall approximation error is guaranteed. This is particularly well seen if, in calculating f^* , we do not sum all the terms in (8.12). If we neglect (say) the highest term $c_n T_n$, then we can definitely say that, because $|T_n| \leq 1$ for all values in $[a, b]$, the resulting error is smaller than $|c_n|$. Such an estimate is impossible in other interpolation procedures.

Naturally the Chebyshev approximation can be used in this form only if we can also calculate the required values of the function for all the given points. However, we want to approximate the coordinates of celestial bodies that, in principle, are available for any arbitrary time. Because of this, this restricted form is suitable for our application.

The evaluation of equation (8.13) in its current form is still computation-intensive. For every expansion coefficient c_j in the second factor of the sum, $(n+1)$ cosine terms have to be calculated. In calculating the n -th order approximation, we are thus forced to evaluate the cosine $(n+1)^2$ times. We shall see if we can find a better solution.

For example, the points $\hat{x}_k^{n+1}$ can be easily calculated recursively. For $k = 1, 2, \dots, (n+1)$ is in fact

$$\hat{x}_k^{n+1} = \cos\left(\frac{\pi}{2n+2}\right), \cos\left(\frac{3\pi}{2n+2}\right), \cos\left(\frac{5\pi}{2n+2}\right), \dots, \cos\left(\frac{(2n+1)\pi}{2n+2}\right).$$

We can obtain this series of values very rapidly by using equation (8.9), if we set $\varphi = \pi/(2n+2)$. This can be rewritten as a program as follows:

```

PHI:=PI/(2*N+2);                      (* h(k)=cos(pi*k/N/2) *)
H[0]:=1.0; H[1]:=COS(PHI);
FOR K:=2 TO (2*N+1) DO H[K]:=2*H[1]*H[K-1]-H[K-2];

```

The array element $H[2*K-1]$ then contains the zero point $\hat{x}_k^{n+1}$. The values

$$\cos(j\pi(2k-1)/(2n+2))$$

can also be calculated in precisely the same way.

Here and later we have to start from a series expansion of lunar positions between times T_a and T_b . The specific points $T[K]$ as well as the lunar coordinates in longitude $L[K]$ and latitude $B[K]$, and the distance $R[K]$ at these times are determined as follows:

```

BMA := (TB-TA)/2.0;                    (* half width of interval *)
BPA := (TA+TB)/2.0;                    (* middle of interval *)
FOR K:=1 TO N+1 DO T[K] := H[2*K-1]*BMA+BPA (* subdivision points *)
FOR K:=1 TO N+1 DO MOON(T[K],L[K],B[K],R[K]); (* value of function *)
FOR K:=2 TO N+1 DO
  IF (L[K-1]<L[K]) THEN L[K]:=L[K]-360.0;

```

The last loop ensures that there are no gaps in the longitudes because of a jump from 360° to 0° . Such an increment must, in any case, be made continuously, in order to avoid erroneous results in the expansion.

In order that we do not have to continually pass individually the interval range, the expansion's order, and the vectors of the polynomial coefficients, it is advisable to define a data structure of the form:

```

CONST MAX_TP_DEG = 13;
TYPE TPOLYNOM = RECORD                      (* Chebyshev polynomial *)
  M : INTEGER;                             (* degree *)
  A,B: REAL;                               (* interval *)
  C : ARRAY [0..MAX_TP_DEG] OF REAL;      (* coefficients *)
END;

```

in the main program. The `T_FIT_MOON` procedure calculates the Chebyshev expansion of the lunar coordinates. The `T_FIT_LBR` sub-routine that is called from it can also be applied to the general expansion of solar and planetary coordinates. Instead of the `POSITION` parameter, `SUN200` or one of the sub-routines `MER200`, ..., `PLU200` from Chap. 6 may then be included.

```

(*-----*)
(* T_FIT_LBR: expands lunar or planetary coordinates into series of *)
(*      Chebyshev polynomials for longitude, latitude and radius *)
(*      that are valid for a specified period of time *)
(*      *)
(* POSITION: routine for calculating the coordinates L,B,R *)
(* TA      : first date of desired period of time *)
(* TB      : last date *)
(* N       : highest order of Chebyshev polynomials (N<=MAX_TP_DEG) *)
(* L_POLY  : coefficients for longitude *)
(* B_POLY  : coefficients for latitude *)
(* R_POLY  : coefficients for radius *)
(*      *)
(* note: *)
(* . the interval [TA,TB] must be shorter than one revolution! *)
(* . the routine will only work for heliocentric planetary or *)
(*      geocentric lunar but not for geocentric planetary coordinates! *)
(*-----*)

```

```

PROCEDURE T_FIT_LBR ( PROCEDURE POSITION (T:REAL; VAR LL,BB,RR: REAL);
                     TA,TB: REAL; N: INTEGER;
                     VAR L_POLY,B_POLY,R_POLY: TPOLYNOM);
CONST PI = 3.1415926535898;
NDIM = 27;
VAR I,J,K      : INTEGER;
    FAC,BPA,BMA,PHI: REAL;
    T,H,L,B,R   : ARRAY[0..NDIM] OF REAL;
BEGIN
  IF (NDIM<2*MAX_TP_DEG+1) THEN WRITELN(' NDIM too small in T_FIT_LBR');
  IF (N>MAX_TP_DEG) THEN WRITELN(' N too large in T_FIT_LBR');
  L_POLY.M := N; B_POLY.M := N; R_POLY.M := N;
  L_POLY.A := TA; B_POLY.A := TA; R_POLY.A := TA;
  L_POLY.B := TB; B_POLY.B := TB; R_POLY.B := TB;
  BMA := (TB-TA)/2.0; BPA := (TB+TA)/2.0;
  FAC := 2.0/(N+1);
  PHI:=PI/(2*N+2); (* h(k)=cos(pi*k/N/2) *)
  H[0]:=1.0; H[1]:=COS(PHI);
  FOR I:=2 TO (2*N+1) DO H[I]:=2*H[1]*H[I-1]-H[I-2];
  FOR K:=1 TO N+1 DO T[K] := H[2*K-1]*BMA+BPA; (* subdivison points *)
  FOR K:=1 TO N+1 DO POSITION(T[K],L[K],B[K],R[K]);
  FOR K := 2 TO N+1 DO (* make L continuous *)
    IF (L[K-1]<L[K]) THEN L[K]:=L[K]-360.0; (* in [-360,+360] ! *)
  FOR J := 0 TO N DO (* calculate Chebyshev *)
    BEGIN (* coefficients C(j) *)
      PHI:=PI/J/(2*N+2); H[1]:=COS(PHI);
      FOR I:=2 TO (2*N+1) DO H[I] := 2*H[1]*H[I-1]-H[I-2];
      L_POLY.C[J]:=0.0; B_POLY.C[J]:=0.0; R_POLY.C[J]:=0.0;
      FOR K:=1 TO N+1 DO
        BEGIN
          L_POLY.C[J] := L_POLY.C[J] + H[2*K-1]*L[K];
          B_POLY.C[J] := B_POLY.C[J] + H[2*K-1]*B[K];
          R_POLY.C[J] := R_POLY.C[J] + H[2*K-1]*R[K];
        END;
    END;
  END;

```

```

      L_POLY.C[J]:=L_POLY.C[J]*FAC; B_POLY.C[J]:=B_POLY.C[J]*FAC;
      R_POLY.C[J]:=R_POLY.C[J]*FAC;
    END;
  END;
  (*-----*)
  (* T_FIT_MOON: approximates the equatorial coordinates          *)
  (*                   of the Moon by Chebyshev expansions for a    *)
  (*                   given period of time of at most one month    *)
  (*-----*)
  (* TA      : first date (in Julian centuries since J2000)      *)
  (* TB      : last date ( TB < TA + 1 month )                    *)
  (* N       : highest order                                       *)
  (* RA_POLY: coefficients for right ascension                     *)
  (* DE_POLY: coefficients for declination                         *)
  (* R_POLY  : coefficients for geocentric distance                *)
  (*-----*)
  PROCEDURE T_FIT_MOON ( TA,TB: REAL; N: INTEGER;
                        VAR RA_POLY,DE_POLY,R_POLY: TPOLYNOM);
  BEGIN
    T_FIT_LBR ( MOONEQU, TA,TB,N, RA_POLY,DE_POLY,R_POLY );
  END;
  (*-----*)

```

In evaluating a given expansion with Chebyshev polynomials, it is not necessary to calculate the polynomials explicitly as well. An algorithm given by Clenshaw, when applied to the recursive relation from (8.10) yields the following rule for evaluating a Chebyshev expansion (8.12) of order n with coefficients $(c_0, c_1, \dots, c_n)$:

- Set $f_{n+1} = 0$ and $f_{n+2} = 0$.
- Calculate with the normalized argument $\hat{x}$ the series

$$f_i = 2\hat{x}f_{i+1} - f_{i+2} + c_i \quad \text{for } i = n, n-1, \dots, 0 \quad .$$

- The required value of the function is then

$$f(x) = (f_0 - f_2)/2 = \hat{x}f_1 - f_2 + c_0/2 \quad .$$

The function T_EVAL evaluates a Chebyshev-expansion using this method:

```

  (*-----*)
  (* T_EVAL: evaluates the approximation of a function by Chebyshev *)
  (*           polynomials of maximum order F.M over the interval [F.A,F.B] *)
  (* F : record containing the Chebyshev coefficients                *)
  (* X : argument                                                    *)
  (*-----*)
  FUNCTION T_EVAL(F: TPOLYNOM; X: REAL): REAL;
  VAR F1,F2,OLD_F1,XX,XX2 : REAL;
      I : INTEGER;
  BEGIN

```

```

IF ( (X<F.A) OR (F.B<X) ) THEN
  BEGIN WRITELN(' T_EVAL : x not within [a,b]'); END;
F1 := 0.0; F2 := 0.0;
XX := (2.0*X-F.A-F.B)/(F.B-F.A); XX2 := 2.0*XX;
FOR I := F.M DOWNT0 1 DO
  BEGIN OLD_F1 := F1; F1 := XX2*F1-F2+F.C[I]; F2 := OLD_F1; END;
  T_EVAL := XX*F1-F2+0.5*F.C[0]
END;
(*-----*)

```

8.4 The LUNA Program

The LUNA program combines the various routines given in this chapter in one small application. It can be used to calculate a lunar ephemeris—i.e., a table of lunar positions—like those found in many yearbooks. The output consists of the apparent equatorial coordinates of the Moon (referred to the equinox of date), its distance in Earth radii, and its horizontal parallax. LUNA expands the lunar coordinates as a series of Chebyshev polynomials over an interval of ten days at a time. This series-expansion can then be easily and rapidly evaluated. Thirteen lunar positions are calculated so that the approximation given by this series can be determined to the required accuracy. If we therefore require lunar coordinates at intervals of only about one day or more, the series expansion causes a certain penalty in increased computation. On the other hand, ephemerides at smaller intervals (as are required for navigation, for example), can be prepared with a considerable gain in the amount of time required.

```

(*-----*)
(*                LUNA                *)
(*                lunar ephemeris      *)
(*                1993/07/01          *)
(*-----*)

```

```
PROGRAM LUNA(INPUT,OUTPUT);
```

```

CONST MAX_TP_DEG = 13;          (* highest order          *)
      T_OVERLAP   = 3.42E-6;    (* 3h in Julian centuries *)
      T_DEVELOP   = 2.737850787E-4; (* 10d in Julian centuries *)

TYPE TPOLYNOM = RECORD          (* Chebyshev polynomial *)
  M : INTEGER;                  (* degree          *)
  A,B: REAL;                    (* interval        *)
  C : ARRAY [0..MAX_TP_DEG] OF REAL; (* coefficients    *)
END;

```

```

VAR RA,DE,R,PAR, MODJD,HOUR : REAL;
    T,DT,T_START,T_END,TA,TB: REAL;
    DAY,MONTH,YEAR,NLINE    : INTEGER;
    RA_POLY,DE_POLY,R_POLY  : TPOLYNOM;

```

```

(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS, ASN, ATN, ATN2, CART, POLAR, DMS, T_EVAL, T_FIT_LBR *)
(* MJD, CALDAT, ECLEQU, NUTEQU *)
(* MOON, MOONEQU, T_FIT_MOON *)
(*-----*)

```

```

(*-----*)
(* GETEPH: reads the period of time for the ephemeris *)
(*-----*)

```

```
PROCEDURE GETEPH(VAR T1,DT,T2:REAL);
```

```

    VAR YEAR,MONTH,DAY: INTEGER;
        HOUR          : REAL;

```

```
BEGIN
```

```

    WRITELN;
    WRITELN('                                LUNA: lunar ephemeris          ');
    WRITELN('                                Version 93/07/01              ');
    WRITELN('                                (c) 1993 Thomas Pfleger, Oliver Montenbruck ');
    WRITELN;
    WRITELN(' Begin and end of the ephemeris: ');
    WRITELN;
    WRITE (' first date (yyyy mm dd hh.hhh) ');
    READLN (YEAR,MONTH,DAY,HOUR);
    T1 := ( MJD(DAY,MONTH,YEAR,HOUR) - 51544.5 ) / 36525.0;
    WRITE (' last date (yyyy mm dd hh.hhh) ');
    READLN (YEAR,MONTH,DAY,HOUR);
    T2 := ( MJD(DAY,MONTH,YEAR,HOUR) - 51544.5 ) / 36525.0;
    WRITE (' step size (dd hh.hh) ');
    READLN (DAY,HOUR);
    DT := ( DAY + HOUR/24.0 ) / 36525.0;
END;

```

```

(*-----*)
(* WRTLBRP: formatted output *)
(*-----*)

```

```
PROCEDURE WRTLBRP (L,B,R,P:REAL);
```

```

    VAR H,M: INTEGER;
        S : REAL;

```

```
BEGIN
```

```

    DMS(L,H,M,S); WRITE (H:5,M:3,S:5:1);
    DMS(B,H,M,S); WRITE (H:5,M:3,S:5:1); WRITE (R:10:3);

```



```

      DMS(P,H,M,S); IF (H>0) THEN M:=M+60; WRITELN (M:6,S:6:2);
END;

(*-----*)

BEGIN  (* main program *)

  GETEPH(T_START,DT,T_END);  (* read desired dates *)

  WRITELN;
  WRITE  ('      Date      ET      RA      Dec      Distance ');
  WRITELN (' Parallax ');
  WRITE  ('          h      h m s      o ' ' " Earth radii');
  WRITELN ('      ' ' " ');

  T := T_START;  TB := T_START;  NLINE := 0;

  WHILE (T<=T_END) DO
    BEGIN
      IF (T>TB-T_OVERLAP) THEN          (* new expansion of the coordinates *)
        BEGIN
          TA := T-T_OVERLAP;  TB := T+T_DEVELOP+T_OVERLAP;
          T_FIT_MOON (TA,TB,MAX_TP_DEG,RA_POLY,DE_POLY,R_POLY);
        END;

      (* date *)
      MODJD := T*36525.0 + 51544.5;  CALDAT (MODJD,DAY,MONTH,YEAR,HOURL);
      WRITE (YEAR:5,'/',MONTH:2,'/',DAY:2,HOURL:5:1);

      (* coordinates *)
      RA := T_EVAL(RA_POLY,T)/15.0;  IF RA<0.0 THEN RA := RA + 24.0;
      DE := T_EVAL(DE_POLY,T);
      R  := T_EVAL(R_POLY, T);
      PAR := ASN(1.0/R);

      (* print coordinates *)
      WRTLBRP (RA,DE,R,PAR);
      NLINE := NLINE + 1;
      IF (NLINE MOD 5) = 0 THEN WRITELN;

      T := T + DT;

    END;

  END.

(*-----*)

```

When LUNA is run, the expansion of the lunar coordinates is first calculated, which takes a few seconds. The coordinates are then displayed in rapid sequence, until a new series expansion (for the subsequent interval) is required.

To illustrate the operation of the program, we will calculate a lunar ephemeris for 1989 January, with a step of two days. LUNA requires only the limiting dates for the ephemeris, in the form: Year, Month, Day, and Hours, together with the step interval in the form: Days and Hours. It must be noted that here (as with calculations of planetary positions), all times are in Ephemeris Time. The input data are shown here in *italics*.

LUNA: lunar ephemeris

Version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Begin and end of the ephemeris:

first date (yyyy mm dd hh.hhh) *1989 1 1 0.0*
 last date (yyyy mm dd hh.hhh) *1989 1 31 0.0*
 step size (dd hh.hh) *2 0.0*

Date	ET	RA			Dec	Distance	Parallax
	h	h	m	s	o ' "	Earth radii	' "
1989/ 1/ 1	0.0	13	5	26.2	-10 42 58.7	63.053	54 31.41
1989/ 1/ 3	0.0	14	38	15.3	-20 23 19.2	61.961	55 29.10
1989/ 1/ 5	0.0	16	26	11.9	-26 51 42.9	60.407	56 54.71
1989/ 1/ 7	0.0	18	27	52.8	-27 43 3.3	58.861	58 24.44
1989/ 1/ 9	0.0	20	29	55.2	-21 44 30.5	57.789	59 29.44
1989/ 1/11	0.0	22	21	11.9	-10 29 51.2	57.442	59 51.00
1989/ 1/13	0.0	0	3	44.1	2 54 49.4	57.757	59 31.44
1989/ 1/15	0.0	1	46	21.7	15 30 4.5	58.481	58 47.18
1989/ 1/17	0.0	3	36	54.1	24 39 14.7	59.389	57 53.29
1989/ 1/19	0.0	5	35	18.6	28 13 6.1	60.375	56 56.57
1989/ 1/21	0.0	7	31	1.7	25 29 44.7	61.406	55 59.16
1989/ 1/23	0.0	9	14	3.9	17 51 26.1	62.408	55 5.27
1989/ 1/25	0.0	10	44	27.6	7 31 36.2	63.194	54 24.13
1989/ 1/27	0.0	12	8	35.2	-3 38 59.4	63.512	54 7.79
1989/ 1/29	0.0	13	34	26.4	-14 17 56.9	63.141	54 26.88
1989/ 1/31	0.0	15	10	2.7	-23 1 46.5	62.014	55 26.26

What is striking here are the high declinations that the Moon reached on approximately 1989 January 7 and 19. In that year the lines of nodes of the lunar orbit lay in such a position that the obliquity of the ecliptic (ca. 23°5') and the inclination of the lunar orbit (ca. 5°1') were superimposed with the same sign, so the declination could exceed 28°. At specific times during that year, therefore, the Moon was particularly high or low in the sky, and the azimuths of its rising and setting points reached extreme values.

9. Solar Eclipses

Approximately every 30 days the Moon turns its unilluminated face towards the Earth at New Moon. An observer looking down on the north pole of the ecliptic would then see the Sun, Moon and Earth in a straight line. Despite this, the shadow cast by the Moon rarely touches the Earth. Because of the inclination of its orbit, at the time of New Moon the Moon is usually above or below the Earth's orbital plane, so its shadow misses the Earth. The Moon crosses the ecliptic on only two days each month, but if one of these dates coincides with New Moon, then the Sun, Moon and Earth are in line, and the Moon's shadow falls on part of the Earth. Overall, such a solar eclipse occurs twice a year. Anyone within the umbra, which is about 100 km across, sees a total eclipse of the Sun. This is a relatively small area, so few persons ever have the opportunity of witnessing a solar eclipse.

If the plane of the Moon's orbit remained fixed in space, all eclipses would take place in two specific months. Because of the regression of the line of nodes of the lunar orbit, the dates of eclipses shift by an average of three weeks each year. The period of 18.6 years taken for one complete regression of the line of nodes is therefore directly reflected in the pattern of solar eclipses (see Fig. 9.1).

9.1 Times of New Moon

In order to obtain an idea of possible solar eclipses in a year, we can first determine the dates of New Moon for every month and then investigate the position of the Moon relative to the ecliptic. The time of New Moon is the moment when the ecliptic longitudes of the Sun and Moon ($\lambda_{\odot}$ and λ_M , respectively) agree, i.e., when the difference $\lambda_M - \lambda_{\odot}$ disappears. The latter, for its part, consists of the difference D between the mean longitudes, plus the difference between the periodic perturbations:

$$\lambda_M - \lambda_{\odot} = D + (\Delta\lambda_M - \Delta\lambda_{\odot}) \quad .$$

The value of the elongation D is given by (8.5):

$$\begin{aligned} D &= D_0 + D_1 \cdot T \\ &= 297^{\circ}85027 + 445267^{\circ}.11135 \cdot T \end{aligned}$$

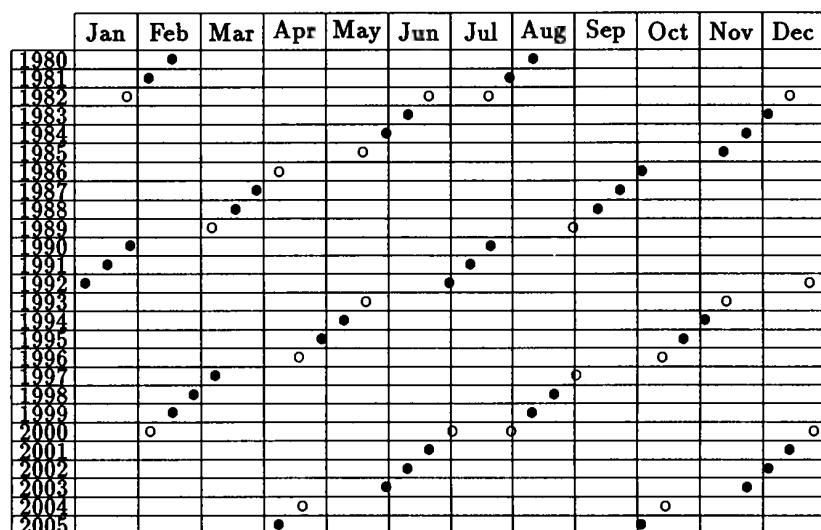


Fig. 9.1. Solar eclipses between 1980 and 2005 (• = total/annular, o = partial)

where

$$T = (JD - 2451545)/36525 \quad .$$

D_1 indicates the amount of change in D over a Julian century. The mean interval between two successive New Moons, i.e., the time in which D alters by 360° , is found to be 29.53 days approximately. This enables the approximate times of New Moon through the year to be calculated very simply.

The periodic perturbations of the lunar and solar orbits vary very little over a short period of time Δt , which means that over that interval $\lambda_M - \lambda_\odot$ essentially varies only by $D_1 \cdot \Delta t/36525^d$. An approximation t_0 for the time of New Moon can therefore be improved by using

$$t_1 = t_0 - \frac{D(t_0) + (\Delta\lambda_M(t_0) - \Delta\lambda_\odot(t_0))}{D_1} \cdot 36525^d \quad .$$

If this step is repeated, then the time of New Moon is determined to a sufficient degree of accuracy. $\Delta\lambda_M$ and $\Delta\lambda_\odot$ may be determined by short series expansions involving the longitudes and anomalies of the Sun and the Moon (cf. (8.2)...(8.5)):

$$\begin{aligned} \Delta\lambda_M &= +22640'' \sin(l) - 4586'' \sin(l - 2D) + 2370'' \sin(2D) \\ &\quad + 769'' \sin(2l) - 668'' \sin(l') + \dots \\ \Delta\lambda_\odot &= +6893'' \sin(l') + 72'' \sin(2l') \quad . \end{aligned}$$

The ecliptic latitude of the Moon can also be expressed by a similar equation:

$$\beta_M \approx +18520'' \sin(F + \Delta\lambda) - 526'' \sin(F - 2D) \quad .$$

The derivation of this equation is given in Chap. 8.

The steps that have been mentioned are combined in the following **NEWMOON** program. After entering the required year, the individual dates of New Moon are determined, together with the corresponding values of the Moon's ecliptic latitude β . The occurrence of solar eclipses can be assessed sufficiently accurately from this information.

Because of the inclination of the Moon's orbit relative to the ecliptic, β varies over the course of the year between -5° and $+5^\circ$. In general, solar eclipses occur at the two dates when New Moon has the (relatively) smallest ecliptic latitude. Nevertheless in some years more than two eclipses may occur. Recent examples are the years 1982 and 2000 with four partial eclipses, each. The greatest ecliptic latitude at which a total eclipse may still occur is around one degree. On the other hand, a partial eclipse may occur up to $\beta \approx 1.5^\circ$. In addition, we can already see approximately where a specific eclipse will be visible. If the Moon is north of the ecliptic ($\beta > 0$), then the area covered by the shadow will lie mainly in the Earth's northern hemisphere.

```
(*-----*)
(*                               NEWMOON                               *)
(*      Date of New Moon and ecliptic latitude of the Moon      *)
(*                               version 93/07/01                               *)
(*-----*)
```

```
PROGRAM NEWMOON (INPUT,OUTPUT);
```

```
(*-----*)
(* The procedures MJD and CALDAT should be entered here *)
(*-----*)
```

```
CONST D1 = +1236.853086; (* rate of change dD/dT of the mean elongation *)
                      (* of the Moon from the Sun (in revol./century) *)
      D0 =  +0.827361; (* mean elongation D of the Moon from the sun *)
                      (* for the epoch J2000 in units of 1rev=360deg *)
```

```
VAR  DAY,MONTH,YEAR,YEAR_CALC : INTEGER;
      HOUR                      : REAL;
      LUNATION_0,LUNATION_I    : INTEGER;
      T_NEW_MOON,MJD_NEW_MOON  : REAL;
      B_MOON                   : REAL;
```

```
(*-----*)
(* IMPROVE: improves an approximation T for the time of New Moon and *)
(* finds the ecliptic longitude B of the Moon for that date *)
(* ( T in julian cent. since J2000, T=(JD-2451545)/36525 ) *)
(*-----*)
```

```
PROCEDURE IMPROVE ( VAR T,B: REAL);
```

```
  CONST P2 =6.283185307; (* 2*pi *)
        ARC=206264.8062; (* arcsec per radian *)
  VAR  L,LS,D,F,DLM,DLS,DLAMBDA: REAL;
```

```

(* with some compilers it may be necessary to replace TRUNC *)
(* by LONG_TRUNC or INT if T<-24! *)
FUNCTION FRAC(X:REAL):REAL;
  BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;
BEGIN
  (* mean elements L,LS,D,F of the lunar orbit *)
  L := P2*FRAC(0.374897+1325.552410*T); (* mean anomaly of the Moon *)
  LS := P2*FRAC(0.993133+ 99.997361*T); (* mean anomaly of the Sun *)
  D := P2*(FRAC(0.5+D0+D1*T)-0.5); (* mean elongation Moon-Sun *)
  F := P2*FRAC(0.259086+1342.227825*T); (* long.Moon-long.asc.node *)
  (* periodic perturbations of the lunar and solar longitude (in") *)
  DLM := + 22640*SIN(L) - 4586*SIN(L-2*D) + 2370*SIN(2*D) + 769*SIN(2*L)
        - 668*SIN(LS) - 412*SIN(2*F) - 212*SIN(2*L-2*D)
        - 206*SIN(L+LS-2*D) + 192*SIN(L+2*D) - 165*SIN(LS-2*D)
        - 125*SIN(D) - 110*SIN(L+LS) + 148*SIN(L-LS) - 55*SIN(2*F-2*D);
  DLS := + 6893*SIN(LS) + 72*SIN(2*LS);
  (* difference of the true longitudes of Moon and Sun in revolutions *)
  DLAMBDA := D / P2 + (DLM - DLS) / 1296000.0;
  (* correction for the time of newmoon *)
  T := T - DLAMBDA / D1;
  (* ecliptic latitude B of the moon (in deg) *)
  B := ( + 18520.0*SIN(F+DLM/ARC) - 526*SIN(F-2*D) ) / 3600.0;
END;
(*-----*)

BEGIN (* main program *)

  WRITELN;
  WRITELN ( ' NEWMOON: Date of New Moon and ecliptic latitude of the Moon' );
  WRITELN ( ' version 93/07/01 ' );
  WRITELN ( ' (c) 1993 Thomas Pfleger, Oliver Montenbruck ' );
  WRITELN;
  WRITE ( ' Dates of New Moon for the year ' ); READLN(YEAR_CALC);
  WRITELN;
  WRITELN ( ' Date ':16,'UT':7,'Latitude':12); WRITELN;
  WRITELN ( ' h':23,'o':9);

  LUNATION_0 := TRUNC( D1 * (YEAR_CALC-2000)/100 );
  FOR LUNATION_I := LUNATION_0 TO LUNATION_0 + 13 DO
    BEGIN
      T_NEW_MOON := ( LUNATION_I - D0 ) / D1;
      IMPROVE ( T_NEW_MOON, B_MOON );
      IMPROVE ( T_NEW_MOON, B_MOON );
      MJD_NEW_MOON := 36525.0*T_NEW_MOON + 51544.5;
      CALDAT ( MJD_NEW_MOON, DAY,MONTH,YEAR,HOURL );
      IF YEAR=YEAR_CALC THEN
        WRITELN(YEAR:12,'/',MONTH:2,'/',DAY:2,HOURL:6:1,B_MOON:9:1)
      END;
    END;
  WRITELN;

END.
(*-----*)

```

9.2 Geometry of an Eclipse

The rays of light leaving the Sun produce two cone-shaped regions, which are known as the umbra and penumbra (Fig. 9.2). For an observer situated in the umbra, the Moon appears larger than the Sun and therefore completely hides the latter.

The point of the umbral cone lies at an average distance of about 375 000 km from the Moon. Farther away, it defines a region in which the solar eclipse appears annular. From here the Moon appears as a dark disk in front of the Sun, which it is unable to cover fully. Part of the Sun remains visible as a bright ring around the Moon. An example of this type of annular solar eclipse is the one that occurred on 1976 April 29, which was observable from part of the Mediterranean Sea.

Within the penumbra, the Sun is only partially eclipsed, and appears as a crescent of greater or lesser width. The percentage that is covered is greater the closer one is to the umbral shadow.

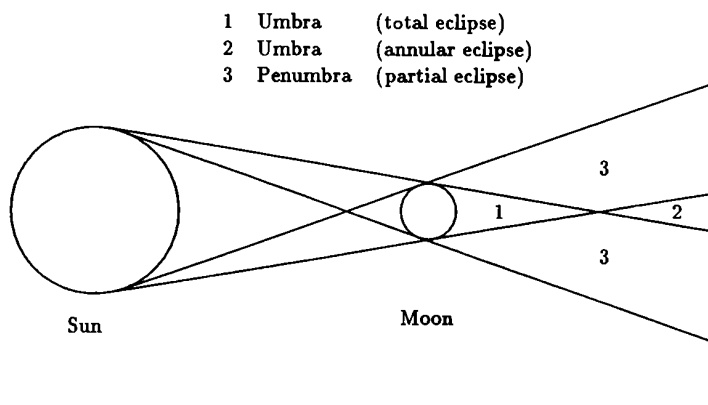


Fig. 9.2. Umbra and Penumbra

The distance of the Moon from the Earth is, on average, 380 000 km, but because of the eccentricity of the lunar orbit, it varies by more than 25 000 km. As a result, however, the Earth is always close to the point of the umbral shadow cone during an eclipse. Even at perigee, when the Earth and Moon are closest, the diameter of the umbral shadow on the surface of the Earth is rarely more than 200 km across. At apogee the eclipse is annular. The borderline case occurs with eclipses that are total for a short period, but which are otherwise annular. An example was the eclipse of 1987 March 29.

Measured in a plane perpendicular to the line joining the Sun and the Moon, and at a distance s behind the Moon, the diameters of the umbra (d)

and of the penumbra (D) are:

$$\begin{aligned} d(s) &= D_{\odot} \left(\frac{s}{r_{\odot M}} \right) - D_M \left(1 + \frac{s}{r_{\odot M}} \right) \quad \text{and} \\ D(s) &= D_{\odot} \left(\frac{s}{r_{\odot M}} \right) + D_M \left(1 + \frac{s}{r_{\odot M}} \right) . \end{aligned} \quad (9.1)$$

Here $D_{\odot}$ and D_M are the diameters of the Sun and Moon respectively:

$$\begin{aligned} D_{\odot} &= 1\,392\,000 \text{ km} = 218.25 R_{\oplus} \\ D_M &= 3\,476 \text{ km} = 0.5450 R_{\oplus} . \end{aligned}$$

$r_{\odot M}$ is the distance between the Sun and the Moon.

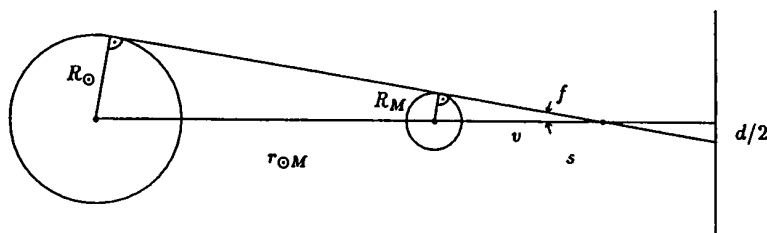


Fig. 9.3. Calculating the diameter of the shadow

The derivation of the equations for the diameter of the shadow is illustrated in Fig. 9.3. There f is half of the apex angle of the shadow cone, and v the distance of the apex from the centre of the Moon. If $R_{\odot}$ and R_M are the radii of the Sun and Moon, respectively, then

$$\sin(f) \cdot v = R_M \quad \text{and} \quad \sin(f) \cdot (v + r_{\odot M}) = R_{\odot}$$

or rearranged

$$\sin(f) = \frac{R_{\odot} - R_M}{r_{\odot M}} \quad \text{and} \quad v = \frac{R_M \cdot r_{\odot M}}{R_{\odot} - R_M} .$$

The diameter of the shadow at a distance s from the Moon is therefore

$$\begin{aligned} d &= 2(s - v) \cdot \tan(f) \\ &\approx 2(s - v) \cdot \sin(f) \\ &= 2R_{\odot} \left(\frac{s}{r_{\odot M}} \right) - 2R_M \left(1 + \frac{s}{r_{\odot M}} \right) . \end{aligned}$$

The equation for the semi-diameter D of the shadow can be derived in a similar way. The sign of d in the above equations is chosen such that in the true shadow, i.e., for a total eclipse, d is *negative*, and is positive for a partial eclipse. Whereas the semi-diameter of the shadow D close to the Earth is approximately half the size of the Earth itself, the diameter d of the umbra varies between $-0.04 R_{\oplus}$ (250 km) and $+0.06 R_{\oplus}$ (350 km).

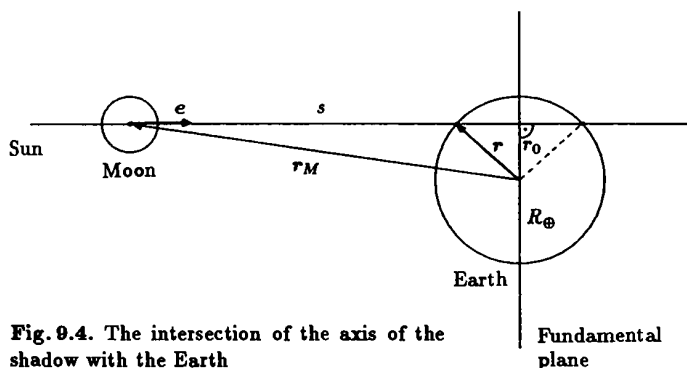


Fig. 9.4. The intersection of the axis of the shadow with the Earth

The position of the umbra on the Earth is obtained if we draw a straight line through the centres of the Sun and Moon, and determine where this intersects the surface of the Earth (see Fig. 9.4). The direction of this straight line is described by the vector

$$\mathbf{e} = \frac{\mathbf{r}_M - \mathbf{r}_\odot}{|\mathbf{r}_M - \mathbf{r}_\odot|} \quad (9.2)$$

of unit length, which may be calculated from the coordinates $\mathbf{r}_\odot$ and $\mathbf{r}_M$ of the Sun and Moon. For a point $\mathbf{r}$ of the shadow we then have

$$\mathbf{r} = \mathbf{r}_M + s\mathbf{e} \quad (9.3)$$

or, expressing this as individual components

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix} + s \cdot \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} .$$

Here s is the distance between the point and the centre of the Moon. Because the tip of $\mathbf{r}$ lies on the surface of the Earth, and if we initially assume the Earth to be a true sphere with radius $R_\oplus$ we have:

$$\mathbf{r}^2 = x^2 + y^2 + z^2 = R_\oplus^2 . \quad (9.4)$$

Combining, we have the quadratic equation

$$s^2 + 2(\mathbf{r}_M \cdot \mathbf{e})s + (r_M^2 - R_\oplus^2) = 0 .$$

This has two solutions

$$s = \begin{cases} s_0 - \sqrt{\Delta} & \text{day side of the Earth} \\ s_0 + \sqrt{\Delta} & \text{night side of the Earth} \end{cases}$$

where

$$\begin{aligned} s_0 &= -\mathbf{r}_M \cdot \mathbf{e} = -(x_M e_x + y_M e_y + z_M e_z) \quad \text{and} \\ \Delta &= s_0^2 + R_\oplus^2 - r_M^2 . \end{aligned} \quad (9.5)$$

s_0 is clearly the distance of the Moon from what is known as the fundamental plane, which is perpendicular to the axis of the shadow and passes through the centre of the Earth. The required point of intersection of the axis of the shadow with the surface of the Earth lies at a distance $\sqrt{\Delta}$ away from this plane on the side that is closest to the Sun and the Moon. The second point of intersection lies on the night side of the Earth, and is of no further relevance for eclipses. Combining the equations, the coordinates of the umbra on the surface of the Earth are given by

$$\mathbf{r} = \mathbf{r}_M + (s_0 - \sqrt{\Delta}) \cdot \mathbf{e} \quad (9.6)$$

We naturally take the discriminant Δ as positive, as otherwise the axis of the shadow would miss the Earth. Nevertheless, a solar eclipse begins before this central phase. As soon as the penumbral shadow touches the Earth, a partial eclipse is observed from somewhere on Earth. If

$$r_0 = \sqrt{r_M^2 - s_0^2}$$

is the distance between the axis of the shadow and the centre of the Earth, and d_0 and D_0 are the diameters of the umbra and the penumbra on the fundamental plane, then we can distinguish between the individual phases of the eclipse by using the following relationships:

$R_\oplus + D_0/2 < r_0$	no eclipse
$R_\oplus + d_0 /2 < r_0 < R_\oplus + D_0/2$	partial phase
$R_\oplus < r_0 < R_\oplus + d_0 /2$	non-central phase
$r_0 < R_\oplus$	central phase

In the non-central phase, the umbral shadow cone brushes the Earth, so that a total or annular eclipse is visible from a small area of the Earth. The axis of the shadow itself, however, does not intersect the surface of the Earth. There is thus no point on the Earth, from which the Moon appears directly in front of the Sun, such that their centres coincide.

These conclusions are based on the assumption that the Earth is a true sphere. In order to take the slight flattening of the Earth into account in calculating eclipses, some small corrections are required. The surface of the Earth can be envisaged as a sphere of radius $R_\oplus$, that is compressed in a direction parallel to its axis by the factor

$$1 - f = 0.996647 \quad .$$

Any point (x, y, z) on this rotational ellipsoid obeys the equation

$$r^2 = x^2 + y^2 + z^2/(1 - f)^2 = R_\oplus^2 \quad , \quad (9.7)$$

which is obtained from (9.4), if we replace z by $z/(1 - f)$. The z -axis of the coordinate system used here is parallel to the Earth's axis (i.e., we are using equatorial coordinates).

Because of the fact that the Earth can be represented as a sphere that is correspondingly stretched, there is a simple method of calculating the position of the umbral shadow correctly. The z coordinates of the Sun and the Moon are multiplied by the factor $1/(1 - f)$, a straight line is laid through these modified positions, and then the point where this intersects a sphere with the radius of the Earth is determined. The only difference between the point of intersection thus calculated and the true point of intersection of the axis of the shadow and the surface of the Earth is that the z coordinate is too great by the factor $1/(1 - f)$. Using this procedure, the relationships determined above can be employed without any alteration.

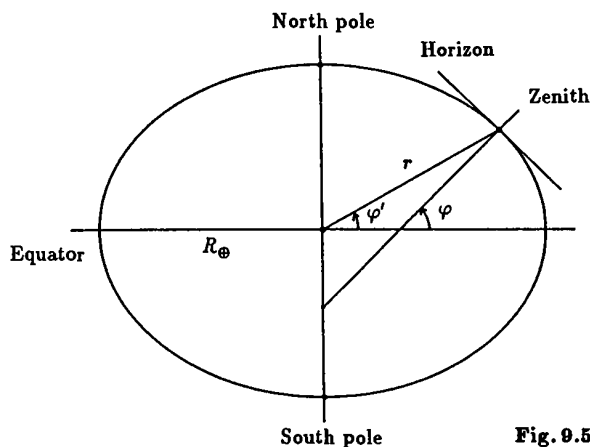


Fig. 9.5. The flattening of the Earth

9.3 Geographic Coordinates and the Flattening of the Earth

Geographic longitude (λ) and latitude (φ) are generally used to describe the coordinates of a point on the surface of the Earth. These are closely related to the equatorial coordinates right ascension (α) and declination (δ), which are employed in astronomy. Both coordinate systems are orientated parallel to the plane of the equator and to the Earth's axis of rotation.

Conversion between declination and geographic latitude would be superfluous if the Earth were a true sphere, because both coordinates would then be identical. Because the Earth is a flattened ellipsoid of rotation – even if only slightly – the concept of geographic latitude must first be refined. φ denotes the angle between the local horizon at any one point and the Earth's axis. This is shown in Fig. 9.5. A cross-section of the Earth through the North and South Poles is an ellipse, to which the horizon is tangent. As can be seen, φ is not identical with geocentric latitude φ' or declination δ , both of which

indicate the angle between the position vector and the equatorial plane. φ and φ' are linked by the equations

$$\begin{aligned} r \cdot \cos(\varphi') &= \frac{R_{\oplus}}{\sqrt{1 - e^2 \sin^2 \varphi}} \cos \varphi \\ r \cdot \sin(\varphi') &= \frac{(1 - e^2) R_{\oplus}}{\sqrt{1 - e^2 \sin^2 \varphi}} \sin \varphi \\ \tan(\varphi') &= (1 - e^2) \tan \varphi \end{aligned} \quad (9.8)$$

Here $R_{\oplus} = 6378.14$ km is the Earth's equatorial radius. e is the eccentricity of the terrestrial ellipsoid. Because of the centrifugal force produced by the Earth's rotation, the Earth is deformed such that the equatorial radius is about 20 km larger than the distance R_{Pol} between the centre and the poles. The relationship between these two values defines the flattening

$$f = \frac{R_{\oplus} - R_{\text{Pol}}}{R_{\oplus}} = \frac{21.385 \text{ km}}{6378.14 \text{ km}} = \frac{1}{298.257} = 0.003353 \quad (9.9)$$

From this the eccentricity of the solid ellipsoid may be derived as

$$e = \sqrt{1 - (1 - f)^2} = \sqrt{2f - f^2} \quad (9.10)$$

Instead of the exact equations (9.8), we can use the perfectly adequate approximation

$$\varphi = \varphi' + 0.1924 \cdot \sin(2\varphi') \quad (9.11)$$

to determine geographic latitude. This clearly indicates that the difference between φ and φ' is a maximum for intermediate latitudes where it amounts to about twelve arc-minutes. At the poles and the equator, the difference vanishes.

The second coordinate used to describe the position of a point on the Earth is geographic longitude. It essentially corresponds to right ascension in an equatorial coordinate system. But whereas right ascension is reckoned from the vernal equinox, a fixed point in space, geographic longitude is measured from the (earth-fixed) Greenwich meridian. The two coordinates therefore differ essentially by an angle $\Theta_0(t)$, which is dependent on the rotation of the Earth in time t :

$$\alpha = \Theta_0(t) + (-\lambda) \quad (9.12)$$

The negative sign of λ takes into account the fact that, in geography, by convention, latitude λ is measured positively towards the West, whereas right ascension α increases in the opposite direction, towards the East. $\Theta_0(t)$ is the corresponding right ascension of the zero meridian, and is therefore simply Greenwich Sidereal Time. The calculation of Sidereal Time has already been described in Section 3.3, where a suitable procedure (LMST) has been given.

An important difficulty that arises here is the necessity for careful discrimination between Universal Time (UT) and Ephemeris Time (ET or TDB/TDT). Although the coordinates of the Sun and Moon must always be calculated only as a function of uniform Ephemeris Time, Universal Time is required as an argument for calculating Sidereal Time. Universal Time cannot, however, be measured with a clock, but, because of its definition, can only be determined from observations. As a result it is only possible to take account retrospectively of the correct difference $\Delta T = ET - UT$ between the two time systems. Some of the measured values for this difference are given in Table (3.1) for the period between 1900 and 1990. In order to avoid having to enter the values every time, we can also resort to using a polynomial approximation of the values given in the table:

$$\begin{aligned}\Delta T &= ET - UT & (9.13) \\ &= +62^{\circ}.14 + 13^{\circ}.34T - 387^{\circ}.70T^2 - 783^{\circ}.42T^3 - 449^{\circ}.50T^4 \\ &\text{with } T = (JD - 2451545)/36525 \text{ and } -1.00 \leq T \leq -0.05\end{aligned}$$

This approximation, which is accurate to about 1-2 s, can only be used for the period between 1900 and 1995. Outside this range it gives incorrect results. The ETMINUT procedure tests for this condition, and if it is not met returns the value FALSE in the variable VALID. Because the value of ET-UT must be determined from observations, it is unfortunately not possible to give long-term predictions. The current data are, however, published in various yearbooks, where they may be found.

```
(*-----*)
(* ETMINUT: Difference of Ephemeris Time and Universal Time *)
(*      (polynomial approximation valid from 1900 to 1995) *)
(*      T:      time in Julian centuries since J2000 *)
(*      ( = (JD-2451545.0)/36525.0 ) *)
(*      DTSEC: DT=ET-UT in sec (only if VALID=TRUE) *)
(*      VALID: TRUE for times between 1900 and 1995, FALSE otherwise *)
(*-----*)
PROCEDURE ETMINUT(T: REAL; VAR DTSEC: REAL; VAR VALID: BOOLEAN);
  BEGIN
    VALID := ( (-1.0<=T) AND (T<=-0.05) );
    IF (VALID) THEN
      BEGIN
        DTSEC := ((((-449.50*T-783.42)*T-387.70)*T)+13.34)*T+62.14;
      END;
    END;
  END;
```

The value of the difference between ET and UT varies very slowly and may, therefore, be regarded as constant for the duration of a solar eclipse.

9.4 Duration of an Eclipse

The duration of the total or annular phase of an eclipse is not the same for every point on the central line. Apart from the diameter of the umbra, it largely depends on the velocity with which the shadow moves across the surface of the Earth. An accurate calculation of the duration of totality requires an iterative determination of the times at which the total phase of the eclipse begins and ends, and is thus computation-intensive. To assess the course of the eclipse and choose a favourable observing site on the central line, however, the following method is completely adequate.

The equatorial coordinates

$$\mathbf{r} = (x, y, z) \quad \text{and} \quad \mathbf{r}' = (x', y', z')$$

of the points at which the axis of the shadow intersects the Earth at times t and $t + \Delta t$ that are close together, may be calculated from equations (9.5) and (9.6), which have already been discussed¹. The difference between these two vectors is a measure of the velocity at which the umbral shadow is moving. Nevertheless, $\mathbf{r}' - \mathbf{r}$ does not give us the required motion of the shadow relative to the surface of the Earth.

The angle w through which the Earth rotates in a time Δt is derived from the duration of a complete rotation, $23^{\text{h}}56^{\text{m}}$, and is

$$w = 2\pi \cdot \frac{\Delta t}{1436^{\text{m}}} \quad (\text{radians}) \quad .$$

For a point on the surface of the Earth that has the equatorial coordinates $\mathbf{r}' = (x', y', z')$ at time $t + \Delta t$, the coordinates at time t are

$$\mathbf{r}'' = \begin{pmatrix} +x' \cdot \cos w + y' \cdot \sin w \\ -x' \cdot \sin w + y' \cdot \cos w \\ +z' \end{pmatrix} \approx \begin{pmatrix} x' + wy' \\ y' - wx' \\ z' \end{pmatrix} \quad ,$$

which are derived from $\mathbf{r}'$ by rotation around the z -axis through angle w . The path that the centre of the umbra traces across the surface of the Earth during the time Δt is therefore

$$\Delta \mathbf{r} = \mathbf{r}'' - \mathbf{r}$$

or

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} x' + wy' & - & x \\ y' - wx' & - & y \\ z' & - & z \end{pmatrix} \quad .$$

¹Because of other approximations that are made here, the flattening of the Earth is ignored in this section.

If $\mathbf{e} = (e_x, e_y, e_z)$ describes the direction of the axis of the shadow, as in (9.3), then $\Delta\mathbf{r}$ may be resolved into one component $\Delta\mathbf{r}_{\parallel}$ parallel to $\mathbf{e}$ and another $\Delta\mathbf{r}_{\perp}$ perpendicular to the axis of the shadow. The lengths of these two components are

$$\Delta\mathbf{r}_{\parallel} = \Delta\mathbf{r} \cdot \mathbf{e} = \Delta x e_x + \Delta y e_y + \Delta z e_z$$

and

$$\Delta\mathbf{r}_{\perp} = \sqrt{(\Delta\mathbf{r})^2 - (\Delta\mathbf{r}_{\parallel})^2} \quad .$$

$\Delta\mathbf{r}_{\perp}$ is the length of the path traced out, perpendicular to the direction of incidence of the shadow, by the umbra's point of contact in time Δt . The duration τ of totality or of the annular phase, for a diameter $|d|$ of the umbral cone, thus equals

$$\tau = \frac{|d|}{\Delta\mathbf{r}_{\perp}} \cdot \Delta t \quad .$$

9.5 Solar and Lunar Coordinates

Accurate coordinates of the Sun and the Moon are naturally important prerequisites for calculating solar eclipses. In order to be able to determine the central line to an accuracy of about 10 km, for example, the position of the Moon in its orbit must be known with comparable accuracy. For an average distance of 380 000 km, this corresponds to an angle of 5". The geocentric solar coordinates must be known with similar accuracy. These requirements are fully met by the **SUN200** and **MOON** procedures, which have already been discussed. The basis of perturbation series that they use need not be examined here. They can be found in Chap. 6 and Chap. 8.

For our purposes, however, it is desirable to use procedures that are not only accurate, but also fast, because of the large number of solar and lunar positions that are required in calculating the course of an eclipse. So the first step is to derive a Chebyshev expansion of the individual coordinates. By this we mean obtaining specific polynomials that express the orbits of the Sun and Moon over a short period of time with sufficient accuracy, but which are significantly simpler to evaluate than complete, universally applicable, series expansions. This method was adopted in calculating lunar ephemerides (Chap.6). The basis for Chebyshev expansions, and the corresponding procedures (**T_EVAL**, **T_FIT_LBR**, and **T_FIT_MOON**) were discussed there, so the reader should already be familiar with them.

The expansion of the lunar coordinates merely requires calling the routine **T_FIT_MOON**. It is assumed that the appropriate data type **TPOLYNOM** has previously been declared. As solar eclipses never last more than a few hours, it is sufficient to use polynomials of order eight for the expansion.


```

(*-----*)
(* T_FIT_SUN: approximates the equatorial coordinates of the      *)
(*              Sun by Chebyshev expansions for a given period of time *)
(*-----*)
(* TA      : first date (in Julian centuries since J2000)        *)
(* TB      : last date ( TB < TA + 1 year )                       *)
(* N       : highest order                                         *)
(* RA_POLY: coefficients for right ascension                      *)
(* DE_POLY: coefficients for declination                          *)
(* R_POLY  : coefficients for geocentric distance                 *)
(*-----*)
PROCEDURE T_FIT_SUN ( TA,TB: REAL; N: INTEGER;
                    VAR RA_POLY,DE_POLY,R_POLY: TPOLYNOM);
BEGIN
  T_FIT_LBR (SUNEQU,TA,TB,N,RA_POLY,DE_POLY,R_POLY);
END;
(*-----*)

```

T_FIT_SUN is used like T_FIT_MOON to calculate the Chebyshev expansion of the solar coordinates. Because of the Sun's very slow motion, a lower-order expansion suffices.

9.6 The ECLIPSE Program

The program ECLIPSE calculates the central line of a solar eclipse and the duration of the total or annular phases, starting with the date of the next New Moon.

The core of the program is the INTSECT sub-routine, in which the intersection of the axis of the shadow with the surface of the Earth is determined from the geocentric coordinates of the Sun and the Moon. In addition, the diameters of the two shadow cones near the Earth are calculated, from which the respective phases of the eclipse are determined. The CENTRAL procedure converts the equatorial coordinates of the umbra provided by INTSECT into geographic coordinates, and also determines the duration of totality.

The program requests the date of New Moon, the difference between Universal Time and Ephemeris Time, and the output step size to be entered. The value $\Delta T = ET - UT$ is, nevertheless, only known for earlier dates, and can otherwise be only approximately estimated. Normally, in predicting eclipses, a value $ET - UT = 0$ is assumed. The coordinates of a point on the central line thus calculated are shifted in geographic longitude relative to the correct ones. This difference corresponds to the rotation of the Earth during the short interval ΔT . If λ' is the calculated longitude, then for the convention adopted here ($\lambda > 0$ towards the West), the actual geographic longitude is:

$$\lambda = \lambda' - 0^{\circ}25^m \cdot \Delta T \quad .$$

For the present-day value of $\Delta T \approx 1^m$, the coordinates predicted by using

$\Delta T = 0$ are thus about $1/4^\circ$ too far West. At the equator this amounts to about 28 km.

```
(*-----*)
(*                               ECLIPSE                               *)
(*           central line and duration of a solar eclipse           *)
(*                               1993/07/01                          *)
(*-----*)
```

```
PROGRAM ECLIPSE(INPUT,OUTPUT);
```

```
CONST MAX_TP_DEG = 8;           (* max. degree of Chebyshev polyn. *)
      H          = 1.14E-6;     (* 1h in jul.cent. (1/(24*36525)) *)
```

```
TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;
      TPOLYNOM = RECORD          (* Chebyshev polynomial *)
        M : INTEGER;             (* Degree *)
        A,B: REAL;               (* Interval *)
        C : ARRAY [0..MAX_TP_DEG] OF REAL; (* Coefficients *)
      END;
      PHASE_TYPE = ( NO_ECLIPSE, PARTIAL,
                     NON_CEN_ANN, NON_CEN_TOT, ANNULAR, TOTAL );
```

```
VAR T_BEGIN,T_END,T,DT,MJDUT      : REAL;
    ETDIFUT                       : REAL;
    LAMBDA,PHI,T_UMBRA            : REAL;
    RAM_POLY,DEM_POLY,RM_POLY     : TPOLYNOM;
    RAS_POLY,DES_POLY,RS_POLY     : TPOLYNOM;
    PHASE                         : PHASE_TYPE;
```

```
(*-----*)
(* The following procedures should be entered here in the given order: *)
(* SN, CS, ASN, ATN, ATN2, CART, POLAR, DMS, T_EVAL, T_FIT_LBR *)
(* ECLEQU *)
(* PMATEQU, PRECART, NUTEQU *)
(* MJD, CALDAT, LMST, ETMINUT *)
(* SUN200, SUNEQU, T_FIT_SUN *)
(* MOON, MOONEQU, T_FIT_MOON *)
(*-----*)
```

```
(*-----*)
(* GET_INPUT: determines search interval, step size and ET-UT *)
(* (T_BEGIN,T_END,DT in Julian centuries since J2000 UT; ET-UT in sec) *)
(*-----*)
```

```
PROCEDURE GET_INPUT ( VAR T_BEGIN, T_END, DT, ETDIFUT: REAL );
```

```
VAR D,M,Y      : INTEGER;
    UT,T,DTAB  : REAL;
    VALID      : BOOLEAN;
```

```
BEGIN
```

```
  WRITELN;
```

```
  WRITELN ( '                               ECLIPSE: solar eclipses           ' );
```

```
  WRITELN ( '                               93/07/01                          ' );
```

```
  WRITELN ( '                               (c) 1993 Thomas Pflieger, Oliver Montenbruck ' );
```

```

WRITELN;
WRITE (' Date of New Moon (yyyy mm dd UT): '); READLN(Y,M,D,UT);
WRITE (' Output step size (min)           : '); READLN(DTAB);
DT := (DTAB/1440.0)/36525.0;
UT := TRUNC ( UT*60.0/DTAB + 0.5 ) * DTAB/60.0 ; (* round to 1 min *)
T := (MJD(D,M,Y,UT)-51544.5)/36525.0;
T_BEGIN := T-0.25/36525.0;  T_END := T+0.25/36525.0;
ETMINUT ( T, ETDIFUT, VALID);
WRITE (' Difference ET-UT (sec)           : ');
IF (VALID) THEN WRITE (' (proposal:',TRUNC(ETDIFUT+0.5):4,' sec) ');
READLN(ETDIFUT);
WRITELN;
WRITELN ('      Date      UT      Phi      Lambda      Durat  Phase ');
WRITELN;
WRITELN ('              h m      o ''      o ''      min      ');
END;

(*-----*)
(* WRTOUT: formatted output *)
(*-----*)

PROCEDURE WRTOUT ( MJDUT,LAMBDA,PHI,T_UMBRA: REAL; PHASE: PHASE_TYPE );
VAR DAY,MONTH,YEAR,H,M: INTEGER;
    HOUR,S : REAL;
BEGIN
  CALDAT ( MJDUT,DAY,MONTH,YEAR,HOUR ); (* date *)
  WRITE ( YEAR:5,'/',MONTH:2,'/',DAY:2);
  DMS(HOUR+0.5/60.0,H,M,S); WRITE (H:4,M:3); (* time rounded to 1 min *)
  IF ( ORD(PHASE)<ORD(ANNULAR) )
    THEN
      WRITE('      -- --      -- --      ---')
    ELSE
      BEGIN
        DMS(PHI,H,M,S);  WRITE (H:9,M:3);
        DMS(LAMBDA,H,M,S); WRITE (H:7,M:3); WRITE (T_UMBRA:7:1);
      END;
  CASE PHASE OF
    NO_ECLIPSE : WRITE('      ----      ');
    PARTIAL    : WRITE('      partial      ');
    NON_CEN_ANN : WRITE('      annular (non-central) ');
    NON_CEN_TOT : WRITE('      total (non-central) ');
    ANNULAR     : WRITE('      annular      ');
    TOTAL       : WRITE('      total      ');
  END;
  WRITELN;
  WRITELN;
END;

(*-----*)
(* INTSECT: calculates the intersection of the shadow axis with the *)
(* surface of the Earth *)
(* *)
(* RAM,DEM,RM, equatorial coordinates of Moon and Sun (right asc. RA *)
(* RAS,DES,RS: and declination in deg; distance in Earth radii) *)

```

```

(*) X,Y,Z:      equatorial coord. of the shadow point (in Earth radii) *)
(*) EX,EY,EZ:   unit vector of the shadow axis                        *)
(*) D_UMBRA:     umbra diameter in Earth radii                        *)
(*) PHASE:       phase of the eclipse                                *)
(*)-----*)

```

```

PROCEDURE INTSECT ( RAM,DEM,RM, RAS,DES,RS: REAL;
                   VAR X,Y,Z, EX,EY,EZ, D_UMBRA: REAL;
                   VAR PHASE: PHASE_TYPE );

```

```

CONST FAC = 0.996633;      (* ratio polar/equatorial Earth radius *)
      D_M = 0.5450;        (* lunar diameter in units of 1 Earth radius *)
      D_S = 218.25;        (* solar diameter in units of 1 Earth radius *)
VAR  XM,YM,ZM, XS,YS,ZS, XMS,YMS,ZMS, RMS: REAL;
      DELTA, RO, SO, S, D_PENUMBRA : REAL;

```

```

BEGIN

```

```

      CART(RM,DEM,RAM,XM,YM,ZM); ZM:=ZM/FAC; (* solar and lunar coordinat. *)
      CART(RS,DES,RAS,XS,YS,ZS); ZS:=ZS/FAC; (* scale z-coordinate *)

```

```

      XMS:=XM-XS; YMS:=YM-YS; ZMS:=ZM-ZS; (* vector Sun -> Moon, *)
      RMS := SQRT(XMS*XMS+YMS*YMS+ZMS*ZMS); (* distance Sun -> Moon *)
      EX:=XMS/RMS; EY:=YMS/RMS; EZ:=ZMS/RMS; (* unit vector Sun -> Moon *)

```

```

      SO := -( XM*EX + YM*EY + ZM*EZ ); (* dist. Moon -> fundamental plane *)
      DELTA := SO*SO+1.0-XM*XM-YM*YM-ZM*ZM;
      RO := SQRT(1.0-DELTA); (* dist.center Earth - shadow axis *)

```

```

      D_UMBRA := (D_S-D_M)*SO/RMS-D_M; (* diameter of pen-/umbra *)
      D_PENUMBRA := (D_S+D_M)*SO/RMS+D_M; (* on the fundamental plane *)

```

```

      (* determine phase and shadow coordinates if required *)
      IF ( RO < 1.0 )

```

```

      THEN (* shadow axis intersects the Earth *)
      BEGIN (* -> total or annular eclipse *)

```

```

          S := SO-SQRT(DELTA);
          D_UMBRA := (D_S-D_M)*(S/RMS)-D_M; (* umbra diameter on Earth *)
          X:=XM+S*EX; Y:=YM+S*EY; Z:=ZM+S*EZ;
          Z:=Z*FAC; (* rescale z-coordinate *)
          IF D_UMBRA>0 THEN PHASE:=ANNULAR ELSE PHASE:=TOTAL;

```

```

      END

```

```

      ELSE

```

```

      IF ( RO < 1.0+0.5*ABS(D_UMBRA) )
      THEN (* non-central eclipse *)
          IF D_UMBRA>0 THEN PHASE:=NON_CEN_ANN ELSE PHASE:=NON_CEN_TOT
      ELSE
          IF ( RO < 1.0+0.5*D_PENUMBRA)
          THEN PHASE := PARTIAL (* partial eclipse *)
          ELSE PHASE := NO_ECLIPSE; (* no eclipse *)

```

```

END;

```

```

(*-----*)
(* CENTRAL: central line, phase and duration of the eclipse *)
(*)
(*) T_UT:          time in Julian centuries since J2000 UT *)
(*) ETDIFUT:       difference Ephemeris Time - Universal Time (in sec) *)
(*) RAM_POLY,DEM_POLY,RM_POLY, RAS_POLY,DES_POLY,RS_POLY: *)
(*)              Chebyshev coefficients for solar and lunar coordinates *)
(*) LAMBDA, PHI:   geographic long. and latit. of the shadow center (deg) *)
(*) T_UMBRA:       duration of the total or annular phase (min) *)
(*) PHASE:         phase of the eclipse *)
(*-----*)
PROCEDURE CENTRAL ( T_UT, ETDIFUT          : REAL;
                   RAM_POLY,DEM_POLY,RM_POLY: TPOLYNOM;
                   RAS_POLY,DES_POLY,RS_POLY: TPOLYNOM;
                   VAR LAMBDA,PHI,T_UMBRA   : REAL;
                   VAR PHASE                 : PHASE_TYPE );

CONST AU          = 23454.78;      (* 1AU in earth radii (149597870/6378.14) *)
DT                = 0.1;          (* small time interval; dt = 0.1 min *)
MPC               = 52596000.0;    (* minutes per Julian century (1440*36525) *)
OMEGA             = 4.3755E-3;    (* angular velocity of the earth (rad/min) *)
VAR RAM,DEM,RM, RAS,DES,RS, RA,DEC,R, DX,DY,DZ,D, MJDUT : REAL;
    T,X,Y,Z,EX,EY,EZ,D_UMBRA, XX,YY,ZZ,EXX,EYY,EZZ,DU, W : REAL;
    PH                                           : PHASE_TYPE;
(* calculate lunar and solar coordinates from Chebyshev coefficients *)
PROCEDURE POSITION ( T : REAL; VAR RAM,DEM,RM, RAS,DES,RS : REAL );
  BEGIN
    RAM:=T_EVAL(RAM_POLY,T); RAS:=T_EVAL(RAS_POLY,T);
    DEM:=T_EVAL(DEM_POLY,T); DES:=T_EVAL(DES_POLY,T);
    RM :=T_EVAL(RM_POLY,T); RS :=T_EVAL(RS_POLY,T); RS:=RS*AU;
  END;

BEGIN

  (* julian centuries since J2000 ET *)
  T := T_UT + ETDIFUT/(86400.0*36525.0);
  (* phase of eclipse and coordinates of the shadow at time T *)
  POSITION ( T, RAM,DEM,RM, RAS,DES,RS );
  INTSECT ( RAM,DEM,RM, RAS,DES,RS, X,Y,Z,EX,EY,EZ, D_UMBRA,PHASE );

  (* for central phase only: geogr. coord. and duration of totality *)
  IF ( ORD(PHASE) < ORD(ANNULAR) )
    THEN BEGIN LAMBDA:=0.0; PHI:=0.0; T_UMBRA:=0.0; END
    ELSE
      BEGIN
        (* geographic coordinates: *)
        MJDUT := 36525.0*T_UT + 51544.5;
        POLAR ( X,Y,Z, R,DEC,RA );
        PHI    := DEC + 0.1924*SN(2.0*DEC);
        LAMBDA := 15.0*LMST(MJDUT,0.0)-RA;
        IF LAMBDA>+180.0 THEN LAMBDA:=LAMBDA-360.0;
        IF LAMBDA<-180.0 THEN LAMBDA:=LAMBDA+360.0;
        (* duration of totality for this place *)

```

```

(* (a) shadow coordinates at time T+DT (or T-DT) *)
POSITION ( T+DT/MPC, RAM,DEM,RM, RAS,DES,RS ); W:=+DT*OMEGA;
INTSECT ( RAM,DEM,RM, RAS,DES,RS, XX,YY,ZZ,EXI,EYY,EZZ, DU, PH );
IF (ORD(PH)<ORD(ANNULAR)) THEN
  BEGIN
    POSITION ( T-DT/MPC, RAM,DEM,RM,RAS,DES,RS); W:=-DT*OMEGA;
    INTSECT (RAM,DEM,RM,RAS,DES,RS, XX,YY,ZZ,EXI,EYY,EZZ, DU,PH);
  END;
(* (b) displacement DX,DY,DZ of the shadow on Earth *)
(* and fraction D perpendicular to the shadow axis *)
DX := XX-X+W*Y;  DY := YY-Y-W*X;  DZ := ZZ-Z;
D := SQRT( DX*DX+DY*DY+DZ*DZ -
           (DX*EX+DY*EY+DZ*EZ)*(DX*EX+DY*EY+DZ*EZ) );
T_UMBRA := DT * ABS(D_UMBRA) / D;
END;

END;

(*-----*)

BEGIN (* main program *)

  (* read search interval *)
  GET_INPUT ( T_BEGIN,T_END, DT, ETDIFUT );

  (* Chebyshev approximations *)
  T_FIT_MOON (T_BEGIN-H,T_END+H,8, RAM_POLY,DEM_POLY, RM_POLY);
  T_FIT_SUN  (T_BEGIN-H,T_END+H,3, RAS_POLY,DES_POLY, RS_POLY);

  (* calculate phase and central line of the eclipse *)
  T := T_BEGIN;
  REPEAT
    CENTRAL ( T,ETDIFUT,
              RAM_POLY,DEM_POLY, RM_POLY, RAS_POLY,DES_POLY, RS_POLY,
              LAMBDA, PHI, T_UMBRA, PHASE );
    IF PHASE<>NO_ECLIPSE THEN
      BEGIN
        MJDUT := 36525.0*T + 51544.5;
        WRTOUT ( MJDUT, LAMBDA, PHI, T_UMBRA, PHASE );
      END;
      T := T+DT;
    UNTIL (T > T_END);

  END.

(*-----*)

```

As an example we will calculate the path of the total eclipse that can be seen in 1999 from central Europe. The central line of this eclipse runs very close to Stuttgart and Munich. First we call **NEWMOON**, in order to determine the dates of New Moon in 1999. The various likely eclipse periods can then be estimated from the ecliptic latitude of the Moon. A rule of thumb is that a

total or annular eclipse is possible when the latitude of the Moon is less than one degree. Partial eclipses can occur if the latitude does not exceed 1°5.

NEWMOON: Date of New Moon and ecliptic latitude of the Moon
version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Dates of New Moon for the year 1999

Date	UT h	Latitude o
1999/ 1/17	15.8	2.2
1999/ 2/16	6.7	-0.5
1999/ 3/17	18.7	-3.0
1999/ 4/16	4.2	-4.6
1999/ 5/15	12.0	-5.0
1999/ 6/13	19.2	-4.0
1999/ 7/13	2.6	-2.1
1999/ 8/11	11.2	0.4
1999/ 9/ 9	22.0	2.9
1999/10/ 9	11.7	4.6
1999/11/ 8	4.0	5.0
1999/12/ 7	22.6	3.9

On 1999 February 16 and August 11 the ecliptic latitude of the Moon at New Moon is extremely small, so this is when eclipses will occur. Closer investigation with ECLIPSE shows that the regions of visibility are:

1999 Feb.16: Southern Indian Ocean, Prince Edward Islands, Iles Crozet, Australia (annular)

1999 Aug.11: North Atlantic, Southern England, Central Europe, Near East, India (total)

The path of the total eclipse of 1999 August 11 should now be determined with ECLIPSE. To do this we enter the date of New Moon just determined, the desired step value in minutes, and a figure for the difference between Ephemeris Time and Universal Time. The approximation for $\Delta T = ET - UT$ that can be calculated by the ETMINUT subroutine is only valid between 1900 and 1995. For the required eclipse we therefore enter an estimated value of 60". All input is shown in *italic* in the example that follows. In order to avoid an unnecessarily long output, some values for times of partial phase are omitted from the following print-out.

ECLIPSE: solar eclipses
93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Date of new moon (yyyy mm dd UT): 1999 8 11 11.2
Output step size (min) : 3.0
Difference ET-UT (sec) : 60.0

Date	UT	Phi	Lambda	Durat	Phase
	h m	° '	° '	min	
1999/ 8/11	8 27	-- --	-- --	---	partial
1999/ 8/11	8 30	-- --	-- --	---	partial
. . .	. .	. .	. .	. .	. .
1999/ 8/11	9 24	-- --	-- --	---	partial
1999/ 8/11	9 27	-- --	-- --	---	partial
1999/ 8/11	9 30	-- --	-- --	---	total (non-central)
1999/ 8/11	9 33	44 51	50 36	1.1	total
1999/ 8/11	9 36	46 23	43 47	1.2	total
1999/ 8/11	9 39	47 24	38 36	1.4	total
1999/ 8/11	9 42	48 9	34 14	1.5	total
1999/ 8/11	9 45	48 44	30 22	1.5	total
. . .	. .	. .	. .	. .	. .
1999/ 8/11	10 15	50 3	3 10	2.1	total
1999/ 8/11	10 18	49 56	1 3	2.2	total
1999/ 8/11	10 21	49 46	0-58	2.2	total
1999/ 8/11	10 24	49 35	-2 56	2.3	total
1999/ 8/11	10 27	49 22	-4 51	2.3	total
1999/ 8/11	10 30	49 7	-6 43	2.3	total
1999/ 8/11	10 33	48 52	-8 31	2.3	total
1999/ 8/11	10 36	48 34	-10 16	2.4	total
1999/ 8/11	10 39	48 16	-11 58	2.4	total
1999/ 8/11	10 42	47 56	-13 38	2.4	total
1999/ 8/11	10 45	47 35	-15 15	2.4	total
1999/ 8/11	10 48	47 12	-16 50	2.4	total
1999/ 8/11	10 51	46 49	-18 22	2.4	total
1999/ 8/11	10 54	46 25	-19 53	2.4	total
1999/ 8/11	10 57	45 59	-21 21	2.4	total
1999/ 8/11	11 0	45 33	-22 48	2.5	total
. . .	. .	. .	. .	. .	. .
1999/ 8/11	12 15	29 8	-57 54	1.6	total
1999/ 8/11	12 18	28 8	-59 53	1.5	total
1999/ 8/11	12 21	27 4	-62 4	1.5	total
1999/ 8/11	12 24	25 55	-64 31	1.4	total
1999/ 8/11	12 27	24 39	-67 19	1.3	total
1999/ 8/11	12 30	23 12	-70 44	1.2	total
1999/ 8/11	12 33	21 24	-75 20	1.0	total
1999/ 8/11	12 36	-- --	-- --	---	total (non-central)
1999/ 8/11	12 39	-- --	-- --	---	partial
1999/ 8/11	12 42	-- --	-- --	---	partial
. . .	. .	. .	. .	. .	. .
1999/ 8/11	13 36	-- --	-- --	---	partial
1999/ 8/11	13 39	-- --	-- --	---	partial

If the phase of the eclipse is total, but not central, the axis of the shadow misses the Earth. But there is still an area on the Earth that lies within the umbral cone. This generally occurs at the beginning and end of an eclipse. Eclipses that are total or annular, but do not appear central anywhere on Earth, because the shadow-cone grazes the Earth tangentially, are very rare.

Eclipses of this sort are not worth observing, because of their short duration, and also because they generally occur at high geographic latitudes.

9.7 Local Circumstances

As well as the position of the central line, information about the local circumstances describing the course of a solar eclipse in greater detail for a given point on the Earth are of considerable significance in the observation of eclipses.

Apart from the maximum phase, the magnitude and the duration, the contact times, i.e., the times at which the limbs of the Sun and Moon touch, are particularly important. First and fourth contact denote the times of entering and leaving the penumbral shadow (i.e., of the beginning and end of the partial phases), and thus the duration of the whole eclipse. Similarly, second and third contact (which are also known as internal contacts) define the extent of the total or annular phase, provided the observer is sited at a point that is crossed by the umbra.

Generally the contact times are supplemented by stating the position angle. At first and fourth contacts, the position angle (which is measured from North through East), indicates the point on the solar limb at which the Moon first touches or finally leaves the disk of the Sun. The position angles of the two internal contacts indicate the positions at which the disk of the Sun finally disappears or first reappears (in a total eclipse), or the points at which it begins and ends as an unbroken ring (in an annular eclipse).

The geometrical requirement for observing first or fourth contact at solar eclipses is that the observer shall be located somewhere on the curved boundary of the Moon's penumbra. Second and third contact, in contrast, occur when the observer is at the boundary of the umbral shadow.

As in our discussion of stellar occultations (see Sect. 10.3), we can best describe the situation by using a system of coordinates based on the fundamental plane, which lies perpendicular to the axis of the shadow, and passes through the centre of the Earth (Fig. 9.4). The x -axis and the y -axis lie in the fundamental plane, and the z -axis indicates the direction from the Moon to the Sun. The x -axis is additionally chosen such that it lies through the intersection of the Earth's equator and the fundamental plane. The unit vectors i , j and k directed along the x -, y -, and z -axes of the coordinate system based on the fundamental plane may then be represented by

$$i = \begin{pmatrix} -k_y/\sqrt{k_x^2 + k_y^2} \\ +k_x/\sqrt{k_x^2 + k_y^2} \\ 0 \end{pmatrix} \quad j = k \times i \quad k = \frac{\mathbf{r}_\odot - \mathbf{r}_M}{|\mathbf{r}_\odot - \mathbf{r}_M|} \quad , \quad (9.14)$$

where the vectors $\mathbf{r}_\odot$ and $\mathbf{r}_M$ denote the equatorial coordinates of the Sun and Moon. By projecting $\mathbf{r}_M$ onto the vector bases i , j and k , we obtain the

coordinates

$$\begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix} = \begin{pmatrix} \mathbf{r}_M \cdot \mathbf{i} \\ \mathbf{r}_M \cdot \mathbf{j} \\ \mathbf{r}_M \cdot \mathbf{k} \end{pmatrix}$$

of the Moon in the fundamental-plane system.

Because, by definition, the line joining the Sun and Moon is parallel to the z -axis, x_M and y_M simultaneous are the x - and y -coordinates of the shadow axis on the fundamental plane. It follows from the orientation of the fundamental plane, moreover, that both the umbral and the penumbral cones intersect the fundamental plane as circles, where, in accordance with (9.2), the radii are given by

$$l = z_M \frac{R_\odot \mp R_M}{r_{\odot M}} \mp R_M \quad (9.15)$$

Here, $R_\odot$ and R_M are the radii of the Sun and Moon respectively, and $r_{\odot M} = |\mathbf{r}_\odot - \mathbf{r}_M|$ is the distance between them. The negative sign applies to the umbra, and the positive sign to the penumbra.

To determine when the observer lies on the boundary of one or other of the shadow zones, the geocentric position of the observer is projected onto the fundamental plane. As shown in Sect. 9.3, the equatorial coordinates of an observer at geocentric longitude λ , geocentric latitude φ' , and distance r from the centre of the Earth, are

$$\mathbf{r}_O = \begin{pmatrix} r \cos(\varphi') \cos(\Theta_0(t) - \lambda) \\ r \cos(\varphi') \sin(\Theta_0(t) - \lambda) \\ r \sin(\varphi') \end{pmatrix}, \quad (9.16)$$

where $\Theta_0(t)$ is Greenwich Sidereal Time. The observer's coordinates (x_O, y_O, z_O) in the system based on the fundamental plane are again similarly obtained by projecting $\mathbf{r}_O$ onto the vector bases $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$. From the height z_O of the observer above the fundamental plane, together with a knowledge of the angular semi-diameter f of the shadow cone and its diameter l on the fundamental plane, we can calculate the radius

$$L = l - z_O \tan f \quad (9.17)$$

of the shadow cone at the observer's position. For partial and annular eclipses, L is always positive, whereas, by convention, the radius of the umbral region in a total eclipse is negative.

A requirement for the observer's position to lie on the shadow cone is therefore that the distance of the observer from the axis of the shadow should be equal to the radius of the shadow cone at the observer's position. This is equivalent to saying that the distance, as expressed by the function

$$f(t) = (x_M - x_O)^2 + (y_M - y_O)^2 - L^2 \quad (9.18)$$

should become equal to zero. The contact times may be determined by iteratively determining the points at which $f(t)$ becomes zero. The time of maximum eclipse may be similarly derived from the point at which $f(t)$ reaches a minimum.

Because the relative positions of the observer and the shadow axis directly reflect the apparent positions of the Sun and Moon, as seen by the observer, the position angle of contacts may be obtained simply from the known coordinates in the fundamental plane. If the position angle of the point of contact is measured from the point due north of the centre of the Sun, positive towards the East, then, irrespective of the type of eclipse, internal and external contacts are given by

$$\begin{pmatrix} \cos P \\ \sin P \end{pmatrix} = \begin{pmatrix} (y_M - y_O)/L \\ (x_M - x_O)/L \end{pmatrix} \quad (9.19)$$

In solving these equations for P , it is important to remember that, as already mentioned, for total eclipses the radius L takes negative values.

Apart from the position angle P , relative to the direction of North, it is also useful to have, for solar eclipses, the position angle relative to the direction of the observer's zenith, $V = P - C$. The parallactic angle C thus corresponds to the position angle of the zenith, and may be calculated sufficiently accurately from the observer's coordinates on the fundamental plane, using

$$\begin{pmatrix} \cos C \\ \sin C \end{pmatrix} \approx \begin{pmatrix} y_O / \sqrt{x_O^2 + y_O^2} \\ x_O / \sqrt{x_O^2 + y_O^2} \end{pmatrix} \quad (9.20)$$

To give specific information about the maximum extent of a solar eclipse, we make use of a quantity, M , known as the magnitude of the eclipse. It is defined as the fraction of the apparent solar diameter that is covered by the Moon at the instant of maximum eclipse. M may be calculated directly from the radii L_1 and L_2 of the penumbral and umbral cones at the observer's position. For total or annular eclipses it is given by

$$M_2 = (L_1 - L_2)/(L_1 + L_2) \quad ,$$

the ratio of the apparent radii of the Sun and Moon. For a partial eclipse we have

$$M_1 = (L_1 - m)/(L_1 + L_2) \quad ,$$

which also depends on the observer's distance m from the axis of the shadow at the time of maximum. It is important to note that the magnitude M is not identical to the eclipsed fraction of the solar disk and, in particular, that at total solar eclipses it assumes values that are greater than one.

9.8 The ECLTIMER Program

Although the track of the central line for a solar eclipse is often known and published years in advance, yearbooks generally contain no details of the local circumstances of an eclipse for a specific observing site. As an extension to the ECLIPSE program we will describe the ECLTIMER program, which enables the local circumstances of an eclipse to be calculated for any given point.

For reasons of space, we will describe just the basic structure of the program, which follows the method of calculation described in the preceding section. The complete, fully commented source code is contained in the file ECLTIMER.PAS on the diskette that accompanies this book.

To determine the contact times, the subroutine CONTACTS is used to calculate the points at which the function, describing the observer's distance from the shadow, $f(t) = m^2(t) - L^2(t)$ (where m is the observer's distance from the axis of the shadow, and L is the radius of the shadow) becomes zero. The shadow distance is calculated by the function SHADOW_DIST, which itself involves the two procedures BESSEL and OBSERVER, which determine the orientation of the fundamental plane, the parameters of the shadow cone, and the observer's coordinates.

By means of quadratic interpolation, we next determine the first and fourth contact, when the observer lies on the penumbral cone. Here we use the interpolation routine QUAD, discussed in Chap. 3, which also enables us to determine when the distance between the observer and the shadow axis is a minimum. Provided we have a partial eclipse (at the very least), CONTACTS calculates the magnitude of the eclipse for the time of maximum.

If, from the distance of the observer from the axis of the shadow and the shadow's diameter at the time of maximum, the eclipse proves to be total or annular, then the times of second and third contacts may be determined. Given that the duration of the total or annular phase can never exceed a value of approximately 13 minutes, second and third contacts must lie within a corresponding interval of time before and after maximum. We can, therefore, employ a procedure similar to regula falsi, which gives a faster result than quadratic interpolation. In ECLTIMER we use a variant of regula falsi, known as the Pegasus method.

Finally, using the POS_ANGLES procedure, the position angles of the internal and external contacts are calculated relative to the direction of North and of the zenith.

When the program is run, ECLTIMER first requests the date of New Moon, and the difference between ET and UT, $\Delta T = ET - UT$. The geographic coordinates of the observing site are then entered. A summary of possible eclipse times may be obtained by using the NEWMOON program.

As an example, we will calculate the local circumstances for the annular eclipse of 1994 May 10. Let us take the observing site as being Rabat in Morocco, with geographical coordinates $\lambda = 6.8333^\circ$ West and $\varphi = 33.95^\circ$.

As an estimated value for the time difference ΔT , we may take a predicted value of 60^s. All input data are shown in *italics* (as usual).

ECLTIMER: Local circumstances of solar eclipses
version 93/07/01

(c) 1993 Thomas Pflieger, Oliver Montenbruck

Date of new moon (YYYY MM DD UT) ... *1994 5 10 17.1*
Difference ET-UT (proposal: 60 sec) ... *60.0*
Observer's coordinates: longitude (>0 west) ... *6.8333*
latitude ... *33.95*

Annular eclipse with $M=0.931$ (0.87). Maximum at 18:58:42 UT.

	h	m	s [UT]	P [°]	V [°]
1st contact: 1994/ 5/10	17	50	47	267	207
2nd contact: 1994/ 5/10	18	56	40	235	179
3rd contact: 1994/ 5/10	19	0	29	120	65
4th contact: 1994/ 5/10	19	59	19	89	38

During the annular phase, which lasts about four minutes for the chosen observing site, the Moon's apparent diameter is about 7% smaller than that of the Sun. In total, only about 87% of the Sun's apparent area is covered by the Moon. Because the eclipse occurs during the evening, the position angles relative to the zenith (V) are about 50°–60° smaller than those relative to North (P). It should also be noted that the Sun sets at Rabat around 19^h18^m UT on the day of the eclipse, so that the end of the eclipse itself cannot be observed.

10. Stellar Occultations

In the course of a day the Moon covers about 13° of its orbit, moving from West to East across the sky. This motion is easiest to see when its orbit takes it close to bright stars. It is particularly striking when a stellar occultation occurs: a star suddenly vanishes behind the eastern limb of the Moon, and then reappears on the other side after a certain interval. In total, there are about one thousand stars visible to the naked eye that the Moon may occult. They lie in a narrow band to the north and south of the ecliptic, and are never more than 8° distant from it.

An occultation is like a solar eclipse in many respects. Because light from the distant star is parallel, however, the Moon's shadow is not conical, but cylindrical with constant diameter. Similarly there is no difference between umbra and penumbra. The shadow of the Moon cast by starlight is as large as the Moon itself ($1/4$ Earth diameter), and is therefore unable to cover the whole Earth. As a result, any one stellar occultation can only be seen from part of the Earth's surface.

The lack of a lunar atmosphere means that a star disappears suddenly behind the Moon, rather than gradually, and that it reappears with equal abruptness. The times of disappearance and reappearance are therefore easy to measure, enabling the position of the Moon to be determined very accurately. The observation of stellar occultations has been used for a long time to improve our knowledge of the motion of the Moon and of the rotation of the Earth.

Basic data for planning the observation of occultations are published in various astronomical yearbooks and journals. The information is, however, restricted to certain specific, major sites. The times of disappearance and reappearance can, however, be calculated for any other observing sites by use of what are known as station coefficients. The calculation of such predictions does not generally require the highest degree of accuracy, so certain simplifications can be made rather than following a rigorous procedure. For example, the irregular profile of the lunar limb can be ignored. Nevertheless, the prediction of stellar occultations is computation-intensive, because a whole series of stars must first be checked to see whether an occultation may occur in the period under consideration.

The OCCULT program calculates possible occultations over a period of ten days for any arbitrary selection of stars. During this period of time, which corresponds to about one third of a lunar orbit, a maximum of one appulse

Table 10.1. Coordinates and proper motions for 100 years (PM) for various stars in the Pleiades. Taken from the *Zodiacal Catalogue*.

No.	Right Ascension (1950)	PM	Declination (1950)	PM	Remarks
	h m s	s	° ' "	"	
536	3 41 49.540	+0.110	+24 8 1.58	-4.26	<i>Celaeno</i>
537	3 41 54.063	+0.144	+23 57 27.90	-4.16	<i>Electra</i>
539	3 42 13.603	+0.171	+24 18 43.07	-4.20	<i>Taygeta</i>
541	3 42 50.749	+0.127	+24 12 46.74	-4.78	<i>Maia</i>
542	3 42 55.387	+0.100	+24 23 59.69	-4.91	<i>Asterope</i>
545	3 43 21.195	+0.145	+23 47 38.98	-4.34	<i>Merope</i>
552	3 44 30.427	+0.144	+23 57 7.52	-4.47	<i>Alcyone</i>
560	3 46 11.022	+0.111	+23 54 7.41	-4.71	<i>Atlas</i>
561	3 46 12.393	+0.114	+23 59 7.18	-5.53	<i>Pleione</i>

of the Moon with a specific star is possible. The minimum separation arises approximately when the geocentric right ascension of the Moon and the star's right ascension are the same. This can be determined with a simple iteration. If this shows that the difference in declination is not too great, then an occultation is possible for at least part of the Earth. Finally, the motion of the lunar shadow around the time of conjunction can be examined to determine whether an occultation will be visible from a given observing site.

10.1 Apparent Positions

The coordinates of a star for which one wishes to calculate details of an occultation can be found in various catalogues. Examples are the *SAO Catalogue* published by the Smithsonian Astrophysical Observatory, and the *Zodiacal Catalogue* (Catalogue of 3539 Zodiacal Stars for the Equinox 1950.0), which was specially prepared for use in determining stellar occultations. A short extract from the *ZC* with the most significant data about various members of the Pleiades is given in Table 10.1. As well as the right ascension and declination, the individual proper motions for one hundred years are given. As can be seen, the spatial motions of stars relative to the Solar System produce marked changes in their coordinates with time. It is therefore necessary first to convert the positions to those prevailing at the approximate time of the occultation by using the given data concerning proper motions.

Even after this correction, however, the positions given in the catalogues mentioned cannot be used directly in calculating occultations. To compare positions with that of the Moon we require what are known as the apparent places of the stars concerned. These are linked to the specific, actual position of the vernal equinox and celestial equator, as governed by precession and

nutations, i.e., to the true equinox of date. The orientation of the coordinate system is always parallel to the rotation axis of the Earth at any moment. The coordinate system in which positions are given in the various stellar catalogues, on the other hand, is defined in terms of the mean equator and vernal equinox for a fixed epoch. In many cases the epoch and equinox are those for 1950, but there are some catalogues that are based on equinox 2000, which was introduced in 1984.

Precession, which describes the mean, long-term changes in the position of the vernal equinox, can be calculated by Equations (2.10) and (2.11). The stellar coordinates for the mean equinox at the epoch given in the catalogues can be converted to the mean equinox of date by using the procedures **PMATEQU** and **PRECART**. Because of the varying gravitational attraction of the Sun and the Moon, an additional periodic variation — nutation — is superimposed on the precession of the Earth's axis. Nutation is described by the two angles $\Delta\epsilon$ and $\Delta\psi$, whose maximum values are $9''$ and $17''$ (cf. (6.16)). A star's true coordinates differ by about this amount from its mean coordinates. The required conversion may be accomplished using Equation (6.17). The routines **PMATEQU** and **NUTEQU** can be combined in a general procedure. **PN_MATRIX** gives a matrix, with which the stellar coordinates for the mean equinox given in a catalogue can be converted directly into coordinates for the true equinox of date. This matrix is defined as an array of type **REAL33**, which must be declared once at the beginning of the program:

```
TYPE REAL33 = ARRAY[1..3,1..3] OF REAL;

(*-----*)
(* PN_MATRIX: combined precession and nutation matrix for transformation *)
(*      from mean equinox T0 to true equinox T *)
(*      T0,T in Julian cent. since J2000; T=(JD-2451545.0)/36525 *)
(*-----*)
PROCEDURE PN_MATRIX ( TO,T:REAL; VAR A: REAL33 );
BEGIN
    PMATEQU(TO,T,A);          (* precession matrix T0->T; *)
    NUTEQU(T,A[1,1],A[2,1],A[3,1]); (* transform column vectors of *)
    NUTEQU(T,A[1,2],A[2,2],A[3,2]); (* matrix A from mean equinox T *)
    NUTEQU(T,A[1,3],A[2,3],A[3,3]); (* to true equinox T *)
END;
(*-----*)
```

The way in which this procedure is used is very similar to that with **PMATEQU** and may be explained by a short example:

```
VAR RA,DEC,R,X,Y,Z,TEQX,T: REAL;
    A : REAL33;
...

TEQX := -0.5;          (* Catalogue equinox 1950 *)
READ (RA,DEC);         (* RA and Dec. in degrees *)
READ (T);              (* Date in Julian cent. since J2000 *)
```

```

PN_MATRIX ( TEQX, T, A );      (* Calculate the Matrix          *)
CART ( 1.0, DEC,RA, X,Y,Z );    (* Cartesian stellar coordinates *)
PRECART ( A , X,Y,Z );          (* multiply by the Matrix      *)
POLAR( X,Y,Z, R,DEC,RA );       (* new polar coordinates RA and DEC *)
WRITELN ( RA, DEC );

```

Another necessary correction has to be applied to stellar coordinates and this is *aberration*, which arises from the finite speed of light. To an observer moving round the Sun with the Earth, the light from a star appears to arrive from a slightly different direction than it would to an observer stationary with respect to the Sun. The observed stellar position can be determined to sufficient accuracy by adding the expression $v_{\oplus}/c$ (the velocity of the Earth relative to that of light) to the rectangular stellar coordinate vector $e = (x, y, z)$. The right ascension and declination of the star, taking aberration into account, may then be obtained from these adjusted Cartesian coordinates. Making the simplifying assumption that the Earth moves in a circle around the Sun, we obtain the velocity of the Earth (expressed in equatorial coordinates) from the three equations

$$\begin{aligned}
 v_{\oplus x}/c &= -0.994 \cdot 10^{-4} \cdot \sin(L) \\
 v_{\oplus y}/c &= +0.912 \cdot 10^{-4} \cdot \cos(L) \\
 v_{\oplus z}/c &= +0.395 \cdot 10^{-4} \cdot \cos(L) \quad ,
 \end{aligned}
 \tag{10.1}$$

in which

$$L = 2\pi \cdot (0.27908 + 100.00214T)$$

describes the heliocentric longitude of the Earth (in radians). As usual, T is the number of Julian centuries since epoch J2000.

```

(*-----*)
(* ABERRAT: velocity vector of the Earth in equatorial coordinates *)
(* (in units of the velocity of light) *)
(*-----*)
PROCEDURE ABERRAT(T: REAL; VAR VX,VY,VZ: REAL);
CONST P2=6.283185307;
VAR L,CL: REAL;
FUNCTION FRAC(X:REAL):REAL;
BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1; FRAC:=X END;
BEGIN
  L := P2*FRAC(0.27908+100.00214*T); CL:=COS(L);
  VX := -0.994E-4*SIN(L); VY := +0.912E-4*CL; VZ := +0.395E-4*CL;
END;
(*-----*)

```

Coordinates taken from a catalogue may be fully corrected taking precession, nutation, and aberration into account, by using the following program segment:

```

VAR RA,DEC,R, X,Y,Z, VX,VY,VZ, TEQX,T: REAL;
  A                                     : REAL33;
...

TEQX := -0.5;                         (* Eqnx. 1950 in Jul. cent. since J2000 *)
READ (RA,DEC);                        (* Right ascension and declination from *)
                                      (* catalogue for mean equinox 1950 *)
READ (T);                             (* Date in Julian. cent. since J2000 *)
PW_MATRIX ( TEQX, T, A );             (* Calculate matrix for Prec.+Nutation *)
ABERRAT ( T, VX,VY,VZ );             (* Velocity of the Earth *)
CART ( 1.0, DEC,RA, X,Y,Z );          (* Cartesian coordinates *)
PRECART ( A , X,Y,Z );               (* multiply by Prec.-Nut.-Matrix *)
X:=X+VX; Y:=Y+VY; Z:= Z+VZ;         (* Aberration *)
POLAR( X,Y,Z, R,DEC,RA );            (* Apparent coordinates RA & Dec *)
WRITELN ( RA, DEC );

```

10.2 Geocentric Conjunction

In order to predict stellar occultations, we have to choose those stars that may be occulted by the Moon in a given period of time, from a large number of stars close to the ecliptic. To do this, we must first determine, for each star, a time at which its right ascension α_* agrees with the geocentric right ascension of the Moon α_M . A comparison of the declinations of the Moon and the star at the time of conjunction allows us to see whether the star will be hidden for at least some portion of the Earth. The time of conjunction also serves as a starting point for the more precise calculation of possible occultations, which will be discussed in later sections.

Between two times t_1 and t_2 the right ascension of the Moon varies between $\alpha_M(t_1)$ and $\alpha_M(t_2)$. If the right ascension of a star α_* lies between these two values, then we can determine the time of conjunction by means of

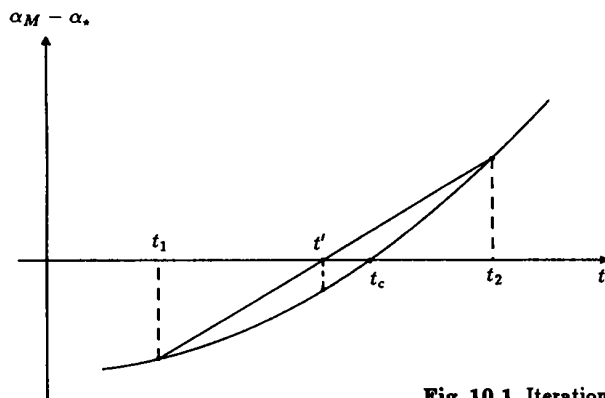


Fig. 10.1. Iteration for the time of conjunction

a linear interpolation (Fig. 10.1). As the right ascension of the Moon increases very evenly, the following close approximation is valid:

$$t_c \approx t' = t_2 - (\alpha_M(t_2) - \alpha_*) \cdot \frac{t_2 - t_1}{\alpha_M(t_2) - \alpha_M(t_1)}.$$

According to whether the right ascension of the Moon at time t' is larger or smaller than that of the star, we replace t_1 or t_2 with t' . Then

$$\alpha_M(t_1) \leq \alpha_* \leq \alpha_M(t_2)$$

is again valid, so we can repeat the procedure to obtain a better time. Because of the regular motion of the Moon, this procedure, known as *regula falsi*, leads rapidly and definitely to the desired result.

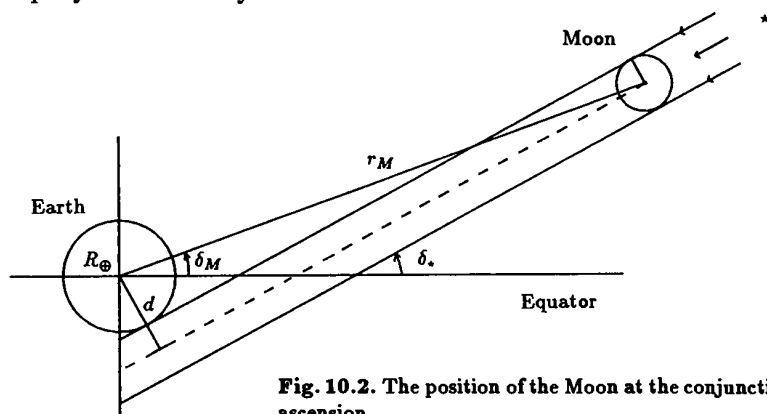


Fig. 10.2. The position of the Moon at the conjunction in right ascension

The possibility of an occultation may now be checked, using the lunar coordinates at the time of conjunction. As may be seen from Fig. 10.2, the shadow cast by the Moon touches the Earth only if the Moon's declination does not differ greatly from that of the star. The distance d between the centre of the Earth and the axis of the shadow should not be greater than the sum of the lunar radius R_M and the Earth's radius $R_\oplus$:

$$d < R_\oplus + R_M \approx 1.3R_\oplus.$$

At the time of conjunction in right ascension, the condition

$$d(t_c) = r_M \cdot |\sin(\delta_M - \delta_*)| < R_\oplus + R_M \approx 1.3R_\oplus$$

must therefore be met. Because of the inclination of the lunar orbit with respect to the celestial equator, $d(t_c)$ can nevertheless be as much as 0.2 Earth radii larger than the smallest limiting distance between the lunar shadow and the centre of the Earth that is otherwise attained. We can therefore select possible occultations by using the following test

$$r_M \cdot |\sin(\delta_M - \delta_*)| \begin{cases} < \\ > \end{cases} 1.5R_\oplus \Rightarrow \begin{cases} \text{occultation possible} \\ \text{no occultation} \end{cases}. \quad (10.2)$$

In searching for a time of conjunction, as well as in other stages of the computation, a whole series of lunar positions therefore has to be determined. It is thus worthwhile employing the expansion of lunar coordinates as Chebyshev polynomials as already discussed in Chap.8. From a few, precisely calculated positions we are able to obtain a simple description of the lunar orbit as a polynomial, which, in just a few operations, may then be evaluated for any number of points in time.

For this we already have the procedure `T_FIT_MOON`, which itself calls the procedures `MOON`, `MOONEQU` and `T_FIT_LBR`. In order to use these, the data type `TPOLYNOM`

```
CONST MAX_TP_DEG = 13;
TYPE TPOLYNOM = RECORD                                (* Chebyshev polynomial *)
    M : INTEGER;                                     (* Order *)
    A,B: REAL;                                       (* Interval *)
    C : ARRAY [0..MAX_TP_DEG] OF REAL; (* Coefficients *)
END;
```

must be included as a global definition. The highest order of the polynomials used ensures that the lunar coordinates can be expressed over a period of about ten days without loss of accuracy. The lines

```
VAR TA,TB: REAL;
    RA_POLY,DE_POLY,
    ...
T_FIT_MOON (TA,TB,MAX_TP_DEG,RA_POLY,DE_POLY,R_POLY);
```

can be used to obtain the coefficients for the approximation of the lunar equatorial coordinates right ascension and declination, as well as the geocentric distance. The corresponding polynomials can be evaluated for any time T within the expansion interval, by means of the procedure `T_EVAL`, which has already been discussed.

```
RA := T_EVAL ( RA_POLY, T ); (* Right ascension in deg. *)
                                (* -360 < =RA <= +360 *)
DEC := T_EVAL ( DE_POLY, T ); (* Declination in deg. *)
R := T_EVAL ( R_POLY, T ); (* Distance in Earth radii *)
```

The coordinates thus determined refer to the true equinox of date, i.e., to the actual positions of the vernal equinox and the celestial equator as affected by precession and nutation.

The first part of our stellar occultation program can now be written. `CONJUNCT` first determines whether the right ascension of the star actually lies within the range of right ascensions covered by the Moon between the two given times T_1 and T_2 . If this is the case, then the time of conjunction is established and a test is carried out to see if the shadow of the Moon touches the Earth. If all the conditions are met, then the time of conjunction is returned to the calling program.

```

(*-----*)
(*  CONJUNCT:                                     *)
(*  *)                                           *)
(*  checks whether there is a conjunction of Moon and star between TA  *)
(*  and TB during which the Moon's shadow hits the Earth                *)
(*  *)                                           *)
(*  TA,TB:    time interval for search of conjunction                    *)
(*  RAPOLY,DEPOLY,RPOLY: Chebyshev coefficients for lunar coordinates  *)
(*  RA,DEC:    right ascension and declination of the star (0<=RA<=360) *)
(*  CONJ:      TRUE/FALSE (conjunction found / no conjunction found)    *)
(*  T_CONJ:    time of conjunction in right ascension (0.0 if CONJ=FALSE) *)
(*  *)                                           *)
(*  All times are counted in Julian centuries since J2000.              *)
(*  The Chebyshev expansion of the lunar right ascension has to yield    *)
(*  values between -360 and +360 degrees and cover less than one orbit. *)
(*-----*)
PROCEDURE CONJUNCT ( TA,TB: REAL; RAPOLY,DEPOLY,RPOLY: TPOLYNOM;
                   RA,DEC: REAL; VAR CONJ: BOOLEAN; VAR T_CONJ: REAL);

CONST EPS=1E-3; (* accuracy in degrees RA *)
VAR  RA_A,RA_B      : REAL;
     T1,T2,T_NEW,DRA1,DRA2,DRA_NEW: REAL;
     DE_CON, R_CON   : REAL;

BEGIN

  T_CONJ := 0.0;
  RA_A := T_EVAL(RAPOLY,TA);
  RA_B := T_EVAL(RAPOLY,TB);
  (* check if RA_A <= RA <= RA_B *)
  CONJ := (RA_A<=RA) AND (RA<=RA_B);
  IF (NOT CONJ) THEN (* check again with RA-360deg *)
    BEGIN RA:=RA-360.0; CONJ := ((RA_A<=RA) AND (RA<=RA_B)); END;
  IF CONJ THEN BEGIN
    (* determine time of conjunction using 'regula falsi' *)
    (* ([T1,T2] always contains T_CONJ) *)
    T1 := TA; DRA1 :=RA_A-RA;
    T2 := TB; DRA2 :=RA_B-RA;
    REPEAT
      T_NEW := T2 - DRA2*(T2-T1)/(DRA2-DRA1);
      DRA_NEW := T_EVAL(RAPOLY,T_NEW) - RA;
      IF DRA_NEW>0 THEN BEGIN T2:=T_NEW; DRA2:=DRA_NEW END
      ELSE BEGIN T1:=T_NEW; DRA1:=DRA_NEW END;
    UNTIL (ABS(DRA_NEW)<EPS);
    T_CONJ := T_NEW;
    (* check if lunar shadow hits the Earth *)
    DE_CON := T_EVAL(DEPOLY,T_CONJ);
    R_CON := T_EVAL(RPOLY, T_CONJ);
    CONJ := ( ABS(SIN(DE_CON-DEC)*R_CON) < 1.5 );
  END;
END;

END;
(*-----*)

```

10.3 The Fundamental Plane

The parallel light from a star casts a cylindrical shadow of the Moon onto the Earth. The shadow appears as a circle on any plane perpendicular to the axis of the shadow, and this has the same diameter as the Moon itself. Such a plane, which also passes through the centre of the Earth, is known as the *fundamental plane*. The relative positions of the Moon's shadow and an observer can be particularly well represented by projecting the positions onto the fundamental plane.

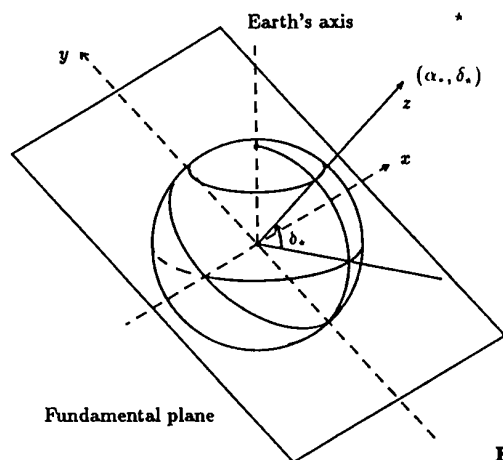


Fig. 10.3. The fundamental plane

If we lay a rectangular coordinate system through the centre of the Earth, as shown in Fig. 10.3, with the z -axis pointing in the direction of the star, which has the coordinates (α_*, δ_*) , and the x -axis in the Earth's equatorial plane, then the x - and y -axes define the fundamental plane. A point at right ascension α and declination δ at a distance r from the Earth then has the following coordinates in this system

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = r \cdot \begin{pmatrix} -\cos \delta \sin(\alpha - \alpha_*) \\ \sin \delta \cos \delta_* - \cos \delta \sin \delta_* \cos(\alpha - \alpha_*) \\ \sin \delta \sin \delta_* + \cos \delta \cos \delta_* \cos(\alpha - \alpha_*) \end{pmatrix}. \quad (10.3)$$

If we substitute the lunar coordinates $(\alpha_M, \delta_M, r_M)$ for (α, δ, r) , then x_M and y_M are the coordinates of the centre of the lunar shadow on the fundamental plane.

In a similar manner, from the equatorial coordinates of the observer, we can obtain the point (x_O, y_O) on the fundamental plane that represents the projection of a ray of light from the star through the observer. The declination of the observing site thus describes the angle between the geocentric position vector and the plane containing the terrestrial and celestial equators. In this

context it is also known as the *geocentric* latitude φ' , which must not be confused with the *geographic* latitude φ . The latter value, which is far more frequently used, is the angle between the Earth's axis and the local horizon, and is, for example, a measure of the altitude of Polaris above the horizon. The flattening of the Earth is the reason for the difference between φ and φ' (cf. Fig. 9.5). Because of its rotation the Earth is not shaped like a sphere, but is instead an ellipsoid of rotation. The distance of the poles from the centre of the Earth is about 20 km less than the Earth's radius at the equator $r_{\oplus} = 6378.14$ km. The ratio of this difference to the Earth's radius is the flattening¹ $f \approx 1/298$. In general the geocentric latitude of a point is not available, and it has to be calculated from the geographic latitude. The latter can be taken from current atlases or reference works. The procedure SITE uses (9.8) to determine the values of $r \cos \varphi'$ and $r \sin \varphi'$ required in (10.3).

```
(*-----*)
(* SITE:  calculates geocentric from geographic coordinates *)
(*      RCPHI:  r * cos(phi') (geocentric; in earth radii) *)
(*      RSPHI:  r * sin(phi') (geocentric; in earth radii) *)
(*      PHI:    geographic latitude (deg) *)
(*-----*)
PROCEDURE SITE ( PHI: REAL; VAR RCPHI,RSPHI: REAL );
  CONST E2 = 0.006694;          (* e**2=f(2-f) for flattening f=1/296.257 *)
  VAR N,SNPHI: REAL;
  BEGIN
    SNPHI := SN(PHI);    N := 1.0/SQRT(1.0-E2*SNPHI*SNPHI);
    RCPHI := N*CS(PHI);  RSPHI := (1.0-E2)*N*SNPHI;
  END;
(*-----*)
```

The right ascension of the observing site is identical to the right ascension of the stars that are on the observer's meridian, and is therefore equal to the local sidereal time. This may be calculated using (3.6) and (3.7). Here we need to consider the different forms of time measurement yet again. In calculating the position of the Moon we have tacitly assumed all the times to be in *Ephemeris Time* ET (= Dynamic Time TDB/TDT). This form of time-measurement, which is physically uniform, is always used in calculating the orbits of celestial bodies. This is basically what the name *Ephemeris Time* implies. But in calculating sidereal time (using the procedure LMST in Sect. 3.3) a knowledge of *Universal Time* is required, which essentially corresponds to ordinary clock time (if time zones are taken into account). The difference between ET and UT currently amounts to about one minute and can be taken from Table 3.1. In addition, a procedure ETMINUT is given in Sect. 9.3 that calculates the desired value for the period between 1900 and 1985 using a polynomial approximation. For later years actual values are given in various astronomical almanacs. As the difference between Universal

¹ f is traditionally used to denote the flattening as well as one of the two coordinates in the fundamental plane. This should not, however, give rise to confusion.

Time and Ephemeris Time alters very slowly, it only has to be determined once at the beginning of our program.

10.4 Disappearance and Reappearance

The actual circumstances of an occultation depend on the differences

$$f = x_M - x_O$$

$$g = y_M - y_O$$

between the coordinates of the Moon and of the observer on the fundamental plane. If the observer's distance $\sqrt{f^2 + g^2}$ from the centre of the lunar shadow is less than the radius of the Moon R_M , then the star will be occulted by the Moon for that particular observer. The times of disappearance and reappearance are governed by the condition

$$f(t)^2 + g(t)^2 = R_M^2 = k^2 \cdot R_\oplus^2 \quad (10.4)$$

Here

$$k = R_M / R_\oplus = 0.2725$$

is the ratio between the radii of the Moon and the Earth.

In its orbit around the Earth, the Moon covers a distance equivalent to the diameter of the Earth in about four hours. During this time the occultation shadow moves across the Earth. In searching for times of contact it therefore suffices to begin about two-and-a-half hours before the time of conjunction. To determine the two times that fulfil condition (10.4), the value of

$$s(t) = f^2(t) + g^2(t) - k^2 R_\oplus^2$$

is calculated for intervals of 15^m. Initially s is positive, because the observing site is still outside the lunar shadow. When s becomes negative, the star disappears behind the limb of the Moon. From three consecutive values s_- , s_0 , and s_+ a parabola may now be determined that closely approximates the changes in $s(t)$ over a period of half an hour. If s reaches zero once or twice during this period, then these points can be obtained by solving a quadratic equation. If not, the next three values for s are taken, and the same calculation is made. This procedure is precisely the same as that used to determine rising and setting times in Chap.3, where the procedure QUAD was also introduced to provide quadratic interpolation and to determine the zero points of a function from three given points. This procedure is employed within SHADOW to determine the times of disappearance and reappearance. SHADOW is given in full at the end of this chapter.

In addition to the times of contact, other data of particular interest to observers are given in predictions of occultations: the position angle of the point

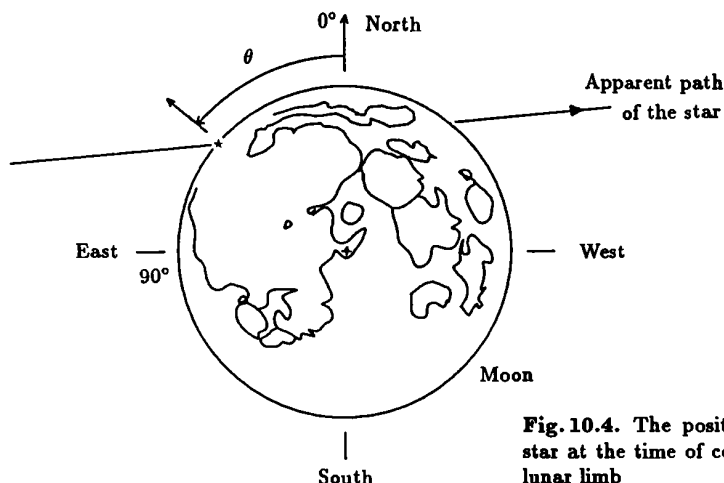


Fig. 10.4. The position angle of a star at the time of contact with the lunar limb

of disappearance and reappearance, and the station coefficients. The definition of the position angle is shown in Fig. 10.4. It is the angle between North and the line from the centre of the Moon through the point of disappearance or reappearance on the lunar limb. Position angles are always measured anticlockwise from 0° – 360° . The value of the position angle ϑ depends only on the fundamental-plane coordinates f and g at the times of disappearance and reappearance, and can be calculated using the relations

$$\begin{aligned}\cos \vartheta &= -g/\sqrt{f^2 + g^2} \\ \sin \vartheta &= -f/\sqrt{f^2 + g^2} .\end{aligned}\quad (10.5)$$

The station coefficients are values that describe an approximate relationship between the contact times and the observing site. They are primarily used to convert the times of contact, predicted for one specific point, to those that will apply at a nearby site. The coefficient a describes how contact time t varies with geographic longitude λ ; b is the corresponding value for geographic latitude φ :

$$a = \frac{dt}{d\lambda} \quad b = \frac{dt}{d\varphi} .$$

Typical values for these two coefficients are around $1^m/^\circ$. For grazing occultations, when the times of disappearance and reappearance lie close together, the station coefficients increase rapidly, however. The equations required to calculate station coefficients mainly contain values that are already known:

$$a = K \cdot [r \cos \varphi' \cdot (f \cos(\alpha_B - \alpha_*) + g \sin \delta_* \sin(\alpha_B - \alpha_*))] \quad (10.6)$$

$$\begin{aligned}b &= K \cdot [r \sin \varphi' \cdot (f \sin(\alpha_B - \alpha_*) - g \cos \delta_* \sin(\alpha_B - \alpha_*)) \\ &\quad - r \cos \varphi' \cdot g \cos \delta_*]\end{aligned}\quad (10.7)$$

where

$$K = -\frac{1^{\text{m}}047/^{\circ}}{(f\dot{f} + g\dot{g})1^{\text{h}}}.$$

Here $\dot{f}$ and $\dot{g}$ are the time derivatives of f and g , which for the sake of simplicity may be expressed as the following differences:

$$\dot{f}(t) \approx \frac{f(t + 0.25^{\text{h}}) - f(t)}{0.25^{\text{h}}} \quad \dot{g}(t) \approx \frac{g(t + 0.25^{\text{h}}) - g(t)}{0.25^{\text{h}}}$$

t is the time of disappearance or reappearance, respectively. The difference between geographic (φ) and geocentric (φ') latitudes can be neglected in calculating the station coefficients.

10.5 The OCCULT Program

The OCCULT program calculates stellar occultations by the Moon over the required period and for a given point. The times of the star's disappearance and reappearance at the lunar limb are calculated together with the corresponding position angles, the altitudes of the Moon above the horizon at the times of contact, and the station coefficients that can be used to convert the times to those that apply at a nearby site.

The coordinates of the stars, whose occultations are being sought, must be entered into a data file called OCCIMP(.DAT). The first line of this file contains the epoch and the equinox of the appropriate catalogue (generally 1950.0 or 2000.0). These data allow for proper motion to be taken into account and for the stellar positions to be converted to the true equinox. Finally each star has a separate line containing

- right ascension (in $^{\text{h}} \text{ } ^{\text{m}} \text{ } ^{\text{s}}$),
- proper motion in right ascension per hundred years (in $^{\circ}$),
- declination (in $^{\circ} \text{ } ' \text{ } ''$),
- proper motion in declination per hundred years (in $''$) and
- the name .

The length of the name is limited to a maximum of 17 letters, but may be increased at any time by altering the constant NAME_LENGTH. The following example of the structure of the data file gives the data for the Pleiades, taken from the *Zodiacal Catalogue* (cf. Table 10.1).

1950.0		1950.0							
3	41	49.540	0.110	24	8	1.58	-4.26	Celaeno	
3	41	54.063	0.144	23	57	27.90	-4.16	Electra	
3	42	13.603	0.171	24	16	43.07	-4.20	Taygeta	

3 42 50.749	0.127	24 12 46.74	-4.78	Maia
3 42 55.387	0.100	24 23 59.69	-4.91	Asterope
3 43 21.195	0.145	23 47 38.98	-4.34	Merope
3 44 30.427	0.144	23 57 7.52	-4.47	Alcyone
3 46 11.022	0.111	23 54 7.41	-4.71	Atlas
3 46 12.393	0.114	23 59 7.18	-5.53	Pleione

At the beginning of the program the geographic coordinates of the observing site are requested, together with the times between which occultations are to be determined. The overall interval is evaluated in steps of ten days, for which the Chebyshev polynomials are first determined. For each star in the data file, after calculating the apparent coordinates (with APPARENT), a check is carried out to see if there is a possible occultation (with the procedure EXAMINE). The CONJUNCT routine, which has already been mentioned, next determines the time of conjunction of the Moon and the star, and checks the distance of the axis of the shadow from the centre of the Earth. SHADOW then determines the times of disappearance and reappearance for the given observing site, as well as the corresponding position angle and station coefficients. However, of the occultations thus determined, an output is generated only for those that occur at least 5° above the horizon and when it is sufficiently dark. DARKNESS tests whether the Sun is at least 6° below the horizon (civil twilight). In contrast to the usual predictions of occultations, no check is made to determine whether the star disappears and reappears at the illuminated or unilluminated limb of the Moon. It must also be mentioned that the output is not necessarily in chronological order of the individual occultations, but instead depends on the order of the entries in the data file.

```
(*-----*)
(*)
(*) OCCULT (*)
(*) prediction of stellar occultations by the Moon (*)
(*) 93/07/01 (*)
(*)
(*)-----*)
```

```
PROGRAM OCCULT(INPUT,OUTPUT,OCCINP);
```

```
CONST MAX_TP_DEG = 13;          (* max. Chebyshev polynom. order *)
      TOVLAP      = 3.42E-6;    (* 3h in Julian centuries *)
      T_SEARCH    = 2.737850787E-4; (* 10d in Julian centuries *)
      NAME_LENGTH = 17;        (* maxim.length of a star's name *)

TYPE REAL33      = ARRAY[1..3,1..3] OF REAL;
      NAME_STRING = ARRAY[1..NAME_LENGTH] OF CHAR;
      TPOLYNOM = RECORD          (* Chebyshev polynomial *)
          M : INTEGER;           (* Order *)
          A,B: REAL;             (* Interval *)
          C : ARRAY [0..MAX_TP_DEG] OF REAL; (* Coefficients *)
      END;
```

```

VAR   T_BEGIN,T_END,T1,T2,TM,T_EQX,T_EPOCH : REAL;
      ETDIFUT                               : REAL;
      RA_STAR,DE_STAR                       : REAL;
      VX,VY,VZ                             : REAL;
      LAMBDA,PHI,RCPHI,RSPHI               : REAL;
      RAPOLY,DEPOLY,RPOLY                  : TPOLYNOM;
      PNMAT                                : REAL33;
      OCCINP                               : TEXT;
      NAME                                 : NAME_STRING;

(*-----*)
(* The following procedures should be entered here in the given order: *)
(* SN, CS, ASN, ATN, ATN2, CART, POLAR *)
(* DDD, DMS, T_EVAL, T_FIT_LBR, QUAD *)
(* ECLEQU, ABERRAT, SITE, PMATEQU, PRECART, NUTEQU, PNMATRIX *)
(* MJD, CALDAT, LMST, ETMINUT *)
(* MOON, MOONEQU, T_FIT_MOON, MINISUN *)
(* CONJUNCT *)
(*-----*)

(*-----*)
(* GET_INPUT: read desired period of time and observer's coordinates *)
(*-----*)
PROCEDURE GET_INPUT(VAR T_BEGIN,T_END,ETDIFUT,LAMBDA,PHI: REAL);
  VAR D,M,Y: INTEGER;
      T      : REAL;
      VALID: BOOLEAN;
BEGIN
  WRITELN;
  WRITELN ( '          OCCULT: occultations of stars by the Moon      ');
  WRITELN ( '                      version 93/07/01                    ');
  WRITELN ( '          (c) 1993 Thomas Pfleger, Oliver Montenbruck    ');
  WRITELN;
  WRITELN ( ' Period of time for prediction of occultations ');
  WRITE ( '   first date (yyyy mm dd)                ... ');
  READLN(Y,M,D);
  T_BEGIN := (MJD(D,M,Y,0)-51544.5)/36525.0;
  WRITE ( '   last date (yyyy mm dd)                  ... ');
  READLN(Y,M,D);
  T_END   := (MJD(D,M,Y,0)-51544.5)/36525.0;
  T := ( T_BEGIN + T_END ) / 2.0;
  ETMINUT ( T, ETDIFUT, VALID);
  IF (VALID)
    THEN WRITE(' Difference ET-UT (proposal:',
              TRUNC(ETDIFUT+0.5):3,' sec)          ... ');
    ELSE WRITE(' Difference ET-UT (sec)              ... ');
  READLN(ETDIFUT);
  WRITE ( ' Observer''s coordinates: longitude (>0 west) ... ');
  READLN(LAMBDA);
  WRITE ( '                      latitude                ... ');
  READLN(PHI);
END;

```

```

(*-----*)
(* HEADER: print header *)
(*-----*)
PROCEDURE HEADER;
  BEGIN
    WRITELN;
    WRITELN ( '   Date      UT      D/R   Pos ',
              '   h      a      b      Star');
    WRITELN ( '           h m s      o',
              '           o      m      m      ');
  END;

(*-----*)
(* GETSTAR: read star coordinates from file OCCINP and correct for *)
(*           proper motion *)
(*-----*)
PROCEDURE GETSTAR ( T_EPOCH,T: REAL;
                   VAR RA,DEC: REAL; VAR NAME: NAME_STRING);
  VAR G,M,I      : INTEGER;
      S, PM_RA, PM_DEC : REAL;
  BEGIN
    READ(OCCINP,G,M,S); DDD(G,M,S,RA); (* right ascension at epoch *)
    READ(OCCINP,PM_RA );
    READ(OCCINP,G,M,S); DDD(G,M,S,DEC); (* declination at epoch *)
    READ(OCCINP,PM_DEC);
    RA := RA + (T-T_EPOCH)*PM_RA /3600.0; (* proper motion right asc. *)
    DEC := DEC + (T-T_EPOCH)*PM_DEC/3600.0; (* proper motion declination*)
    FOR I:=1 TO NAME_LENGTH DO (* name of the star *)
      IF (NOT EOLN(OCCINP) ) THEN READ(OCCINP,NAME[I]) ELSE NAME[I]:=' ';
    READLN(OCCINP);
    RA := 15.0 * RA; (* RA in deg *)
  END;

(*-----*)
(* APPARENT: apparent coordinates of a star *)
(* PNMAT : matrix for precession and nutation *)
(* VX,VY,VZ: velocity of the earth (equatorial coord.; in units of c) *)
(* RA,DEC : right ascension and declination *)
(*-----*)
PROCEDURE APPARENT ( PNMAT: REAL33; VX,VY,VZ: REAL; VAR RA,DEC: REAL );
  VAR X,Y,Z,R: REAL;
  BEGIN
    CART ( 1.0, DEC,RA, X,Y,Z ); (* cartesian coordinates of the star *)
    PRECART ( PNMAT , X,Y,Z ); (* correct for precession and nutation *)
    X:=X+VX; Y:=Y+VY; Z:= Z+VZ; (* aberration *)
    POLAR ( X,Y,Z, R,DEC,RA ); (* apparent right ascension,declination *)
  END;

(*-----*)
(* SHADOW: *)
(* *)
(* starting from the time of conjunction the times, position angles *)
(* and longitude and latitude coefficients of disappearance and *)

```

```

(*) reappearance are calculated for a specific observing site      *)
(*)
(*) RAPOLY,DEPOLY,RPOLY : Chebyshev approximations of lunar coordinates *)
(*) T_CONJ_ET           : time of conjunction in right ascension      *)
(*)                     (in Julian centuries ET since J2000)          *)
(*) ETDIFUT            : ET-UT in sec                                  *)
(*) LAMBDA,RCPHI,RSPHI : geocentric coordinates of the observer      *)
(*) RA_STAR,DE_STAR    : right ascension and declination of the star  *)
(*) EVENT              : TRUE occultation takes place; FALSE otherwise *)
(*) MJD_UT_IN, MJD_UT_OUT: times of contact (Modified Julian Date UT) *)
(*) POS_IN, POS_OUT     : position angles                             *)
(*) H_IN, H_OUT        : star's altitude above the horizon           *)
(*) A_IN, A_OUT        : longitude coefficient                       *)
(*) B_IN, B_OUT        : latitude coefficient                        *)
(*)-----*)

```

PROCEDURE SHADOW

```

  ( RAPOLY, DEPOLY, RPOLY           : TPOLYNOM;
    T_CONJ_ET,ETDIFUT, LAMBDA,RCPHI,RSPHI, RA_STAR,DE_STAR: REAL;
    VAR EVENT                       : BOOLEAN;
    VAR MJD_UT_IN, MJD_UT_OUT       : REAL;
    VAR POS_IN,POS_OUT, H_IN,H_OUT, A_IN,A_OUT, B_IN,B_OUT : REAL      );

CONST DTAB      = 0.25;  (* search step size in hours                *)
RANGE          = 2.25;  (* search interval = +/- (RANGE+DTAB) in hours *)
K              = 0.2725;  (* ratio earth radius / lunar radius *)
CENT           = 876600.0;  (* hours per Julian century          *)
SID            = 1.0027379;  (* ratio solar time / sidereal time *)

```

```

VAR I, NZ, NFOUND      : INTEGER;
K_SQR, MJD_CONJ_UT, HOUR, F, G : REAL;
THETA_CONJ, CSDEST, SNDEST : REAL;
S_MINUS, S_0, S_PLUS, XE, YE : REAL;
Z, TIME                : ARRAY[1..2] OF REAL;

```

(* FG: f-g coordinates in the fundamental plane *)

```

PROCEDURE FG ( HOUR: REAL; VAR F,G: REAL);

```

```

  VAR T,DEM,RM,RCDEM,RSDEM,DRAM,DRA: REAL;

```

```

  BEGIN

```

```

    T := T_CONJ_ET + HOUR/CENT;
    DEM := T_EVAL(DEPOLY,T); RM := T_EVAL(RPOLY,T);
    RCDEM := RM * CS(DEM); RSDEM := RM * SN(DEM);
    DRAM := T_EVAL(RAPOLY,T) - RA_STAR;
    DRA := 15.0 * ( THETA_CONJ + HOUR*SID ) - RA_STAR;
    F := +RCDEM*SN(DRAM) - RCPHI*SN(DRA);
    G := + RSDEM*CSDEST - RCDEM*SNDEST*CS(DRAM)
        - RSPHI*CSDEST + RCPHI*SNDEST*CS(DRA);

```

```

  END;

```

(* CONTACT: position angle, altitude and longitude/latitude coeffic. *)

```

PROCEDURE CONTACT ( HOUR: REAL; VAR POS,H,A,B: REAL );

```

```

  VAR F,G,FF,GG,DF,DG,FAC,DRA,CDRA,SDRA: REAL;

```

```

  BEGIN

```

```

FG ( HOUR, F, G ); FG ( HOUR+DTAB , FF, GG );
DF := (FF-F)/DTAB; DG := (GG-G)/DTAB;
POS := ATN2(-F,-G); IF POS<0.0 THEN POS:=POS+360.0;
FAC := 1.047 / (F*DF+G*DG);
DRA := 15.0 * ( THETA_CONJ + HOUR*SID ) - RA_STAR;
CDRA := CS(DRA); SDRS := SN(DRA);
A := -FAC * RCPHI * ( F*CDRA + G*SDRA*SNDEST );
B := -FAC*( RSPHI * (F*SDRA-G*SNDEST*CDRA) - RCPHI*G*CSDEST );
H := ASN ( RSPHI*SNDEST + RCPHI*CSDEST*CDRA );
END;

```

```
BEGIN
```

```

(* modified julian date and sidereal time at time of conjunction *)
MJD_CONJ_UT := T_CONJ_ET*36525.0 + 51544.5 - ETDIFUT/66400.0;
THETA_CONJ := LMST ( MJD_CONJ_UT, LAMBDA );

```

```
(* auxiliary values *)
```

```
K_SQR := K*K; CSDEST := CS(DE_STAR); SNDEST := SN(DE_STAR);
```

```
(* search for time of contact *)
```

```
NFOUND := 0; TIME[1]:=0.0; TIME[2]:=0.0;
```

```
HOURL := -RANGE-2.0*DTAB;
```

```
FG (-RANGE-DTAB,F,G); S_PLUS := F*F+G*G-K_SQR;
```

```
REPEAT
```

```
HOURL := HOURL + 2.0*DTAB;
```

```
S_MINUS := S_PLUS;
```

```
FG ( HOURL ,F,G ); S_0 := F*F+G*G-K_SQR;
```

```
FG ( HOURL+DTAB,F,G ); S_PLUS := F*F+G*G-K_SQR;
```

```
QUAD ( S_MINUS,S_0,S_PLUS, XE,YE, Z[1],Z[2],NZ );
```

```
FOR I:=1 TO NZ DO TIME[NFOUND+I] := HOURL+DTAB*Z[I];
```

```
NFOUND := NFOUND + NZ; EVENT := (NFOUND=2);
```

```
UNTIL ( (EVENT) OR (HOURL>=RANGE) );
```

```
(* calculate details of an occultation *)
```

```
IF EVENT THEN
```

```
BEGIN
```

```
MJD_UT_IN := MJD_CONJ_UT + TIME[1] / 24.0;
```

```
MJD_UT_OUT := MJD_CONJ_UT + TIME[2] / 24.0;
```

```
CONTACT ( TIME[1], POS_IN, H_IN, A_IN, B_IN );
```

```
CONTACT ( TIME[2], POS_OUT,H_OUT,A_OUT,B_OUT );
```

```
END;
```

```
END;
```

```

(*-----*)
(* DARKNESS: test for civil twilight *)
(* MODJD: Modified Julian Date *)
(* LAMBDA: geographic longitude (>0 west of Greenwich) *)
(* CPHI,SPHI: sine and cosine of the geographic latitude *)
(*-----*)

```

```
FUNCTION DARKNESS ( MODJD, LAMBDA,CPHI,SPHI: REAL ): BOOLEAN;
```

```
VAR T,RA,DEC,TAU,SIN_HSUN: REAL;
```



```

BEGIN
  T := (MODJD-51544.5)/36525.0;
  MINI_SUN (T,RA,DEC);
  TAU := 15.0 * (LMST(MODJD,LAMBDA) - RA);
  SIN_HSUN := SPHI*SN(DEC) + CPHI*CS(DEC)*CS(TAU);
  DARKNESS := ( SIN_HSUN < -0.10 );
END;

(*-----*)
(* EXAMINE: *)
(* checks whether an occultation takes place, calculates the *)
(* circumstances and prints the results *)
(* *)
(* T1,T2 : search interval in Julian cent. since J2000 *)
(* RAPOLY,DEPOLY,RPOLY : Chebyshev approximations of lunar coordinates *)
(* ETDIFUT : ET-UT in sec *)
(* LAMBDA,RCPHI,RSPhi : geocentric coordinates of the observer *)
(* RA_STAR,DE_STAR : star's coordinates *)
(* NAME : star's name *)
(*-----*)

PROCEDURE EXAMINE ( T1,T2: REAL; RAPOLY,DEPOLY,RPOLY: TPOLYNOM;
                   ETDIFUT,LAMBDA,RCPHI,RSPhi,RA_STAR,DE_STAR: REAL;
                   NAME: NAME_STRING);

CONST H_MIN=5.0; (* minimum altitude above the horizon (deg) *)

VAR DAY,MONTH,YEAR,H,M,I : INTEGER;
    S,HOUR, T_CONJ_ET, MJD_UT_IN, MJD_UT_OUT : REAL;
    POS_IN,POS_OUT, H_IN,H_OUT, A_IN,A_OUT, B_IN,B_OUT : REAL;
    CONJ, TAKES_PLACE : BOOLEAN;

BEGIN
  (* test for conjunction in RA and find time of conjunction *)

  CONJUNCT ( T1,T2, RAPOLY,DEPOLY,RPOLY,
             RA_STAR,DE_STAR, CONJ, T_CONJ_ET );

  IF CONJ THEN

    BEGIN
      (* check a possible occultation for the given observing site *)
      (* and calculate times of contact, altitudes and longitude and *)
      (* latitude coefficients *)

      SHADOW ( RAPOLY,DEPOLY,RPOLY, T_CONJ_ET, ETDIFUT,
               LAMBDA,RCPHI,RSPhi, RA_STAR, DE_STAR, TAKES_PLACE,
               MJD_UT_IN, MJD_UT_OUT, POS_IN,POS_OUT,
               H_IN,H_OUT, A_IN,A_OUT, B_IN,B_OUT );

      (* print results if the occultation takes place during the *)

```

```

(* night and high enough above the horizon *)

IF TAKES_PLACE THEN
IF ( (H_IN>H_MIN) OR (H_OUT>H_MIN) ) THEN
IF DARKNESS ( (MJD_UT_IN+MJD_UT_OUT)/2.0, LAMBDA, RCPHI, RSPHI )
THEN

BEGIN

    (* disappearance *)
    CALDAT ( MJD_UT_IN, DAY, MONTH, YEAR, HOUR ); DMS (HOUR, H, M, S);
    WRITE ( (YEAR MOD 100):3, '/',
            MONTH:2, '/', DAY:2, H:5, M:3,
            TRUNC(S+0.5):3, ' D ', TRUNC(POS_IN+0.5):5,
            TRUNC(H_IN+0.5):6, A_IN:8:1, B_IN:6:1, ' ':3 );
    FOR I:=1 TO NAME_LENGTH DO WRITE(NAME[I]); WRITELN;

    (* reappearance *)
    CALDAT ( MJD_UT_OUT, DAY, MONTH, YEAR, HOUR ); DMS (HOUR, H, M, S);
    WRITELN ( (YEAR MOD 100):3, '/',
            MONTH:2, '/', DAY:2, H:5, M:3,
            TRUNC(S+0.5):3, ' R ', TRUNC(POS_OUT+0.5):5,
            TRUNC(H_OUT+0.5):6, A_OUT:6:1, B_OUT:6:1 );

END;

END;

END;

(*-----*)
BEGIN (* main program *)

    (* read search interval and geographic coordinates *)
    GET_INPUT ( T_BEGIN, T_END, ETDIFUT, LAMBDA, PHI );

    (* calculate geocentric coordinates of the observer *)
    SITE ( PHI, RCPHI, RSPHI );

    (* search occultations in subsequent time intervals *)
    T2 := T_BEGIN;

    REPEAT

        T1:=T2; T2:=T1+T_SEARCH;

        (* approximate lunar coordinates by Chebyshev polynomials *)
        T_FIT_MOON ( T1-TOVLAP, T2+TOVLAP, MAX_TP_DEG, RAPOLY, DEPOLY, RPOLY );

```

```

(* print header *)

HEADER;

(* open star catalogue file, read epoch and equinox *)

(* RESET ( OCCINP ); *) (* Standard Pascal *)
ASSIGN (OCCINP,'OCCINP.DAT'); RESET(OCCINP); (* Turbo Pascal *)
(* RESET ( OCCINP,'OCCINP.DAT'); *) (* ST Pascal plus *)

READLN ( OCCINP, T_EPOCH, T_EQX );
T_EQX := ( T_EQX -2000.0 ) / 100.0;
T_EPOCH := ( T_EPOCH-2000.0 ) / 100.0;

(* calculate transformation matrix between the mean equinox of the *)
(* star catalog and the true equinox of the search interval center *)

TM := (T1+T2)/2.0;
PN_MATRIX ( T_EQX, TM, PNMAT );

(* heliocentric velocity of the earth for calculation of aberration *)

ABERRAT ( TM, VX,VY,VZ );

(* loop through list of stars and search for possible occultations *)

WHILE NOT EOF(OCCINP) DO
  BEGIN
    (* read new star coordinates *)
    GETSTAR ( T_EPOCH, TM, RA_STAR,DE_STAR,NAME );
    (* calculate apparent coordinates *)
    APPARENT ( PNMAT, VX,VY,VZ, RA_STAR,DE_STAR );
    (* check for occultation *)
    EXAMINE ( T1,T2, RAPOLY,DEPOLY,RPOLY,
              ETDIFUT, LAMBDA,RCPHI,RSPHI, RA_STAR,DE_STAR,NAME );
  END;
WRITELN;

UNTIL (T2 >= T_END);

END.

(*-----*)

```

As an example, we will now use OCCULT to calculate two series of Pleiades occultations that occurred in 1989. As search period we will take the interval between 1989 September 15 and November 15. The predictions are for Munich, which is at longitude 11°6 East, and latitude 48°1 North. We will assume that the future difference between Ephemeris Time and Universal Time is unknown (as it would be in making predictions), and will estimate the amount to be allowed for this, choosing 56 seconds.

The overall period is searched in steps of ten days, and all occultations discovered are output. In the following list, the output of header lines is suppressed when no occultations of the Pleiades occur in a particular interval. Data entered by the user are in italics.

OCCULT: occultations of stars by the Moon
version 93/07/01

(c) 1993 Thomas Pfleger, Oliver Montenbruck

Period of time for prediction of occultations

first date (yyyy mm dd) ... 1989 09 15
last date (yyyy mm dd) ... 1989 11 15
Difference ET-UT (proposal: 58 sec) ... 56.00
Observer's coordinates: longitude (>0 west) ... -11.60
latitude ... 48.10

Date	UT	D/R	Pos	h	a	b	Star
	h m s		o	o	m	m	
89/ 9/19	21 49 45	D	96	25	-0.3	1.3	Celaeno
69/ 9/19	22 41 3	R	222	34	-0.0	2.1	
89/ 9/19	22 1 60	D	67	27	-0.1	1.7	Taygeta
89/ 9/19	23 0 40	R	251	37	-0.5	1.7	
89/ 9/19	22 15 35	D	98	30	-0.5	1.3	Maia
89/ 9/19	23 7 34	R	220	38	-0.1	2.3	
89/ 9/19	22 21 1	D	62	31	-0.1	1.8	Asterope
89/ 9/19	23 20 38	R	256	41	-0.6	1.6	
...	...						

Date	UT	D/R	Pos	h	a	b	Star
	h m s		o	o	m	m	
89/11/13	18 3 47	D	36	24	0.3	2.0	Celaeno
89/11/13	18 48 58	R	285	31	-0.7	1.1	
89/11/13	17 54 51	D	73	22	0.0	1.5	Electra
89/11/13	18 49 35	R	248	31	-0.2	1.7	
89/11/13	18 27 22	D	37	28	0.2	2.1	Maia
89/11/13	19 14 37	R	284	35	-0.8	1.1	
89/11/13	18 43 45	D	153	30	-5.0	-4.8	Merope
89/11/13	18 50 46	R	168	31	4.2	7.7	
89/11/13	19 0 9	D	122	33	-1.1	0.6	Alcyone
89/11/13	19 36 7	R	198	39	0.4	2.9	

10.6 Estimation of $\Delta T = ET - UT$ from Observations

After observing a stellar occultation, it requires little effort to estimate the difference between Universal Time and Ephemeris Time and thus keep track of the decrease in the Earth's rate of rotation.

To do this, we compare the time of the observed appearance or disappearance at the lunar limb with the value $t_{ET=UT}$ predicted on the assumption that $\Delta T = ET - UT = 0$. Provided the time of observation t_{UT} is referred to

Universal Time, the following relation holds for the same instant expressed in Ephemeris Time:

$$t_{ET} = t_{UT} + \Delta T \quad .$$

The Ephemeris Time t_{ET} of the occultation, however, is not the same as the predicted value $t_{ET=UT}$. This is because the computed Greenwich sidereal time differs from the actual value by $\Delta\theta_0 \approx \Delta T$, when assuming $ET = UT$ in the prediction. Therefore the computed occultation refers to a point on Earth that is located $\Delta\lambda = 15^\circ/h \cdot \Delta T$ to the east of the real observing site. This difference may, however, easily be accounted for by the station coefficient a , yielding

$$t_{ET} = t_{ET=UT} + a(0^\circ 25/m \cdot \Delta T) \quad .$$

After inserting and rearranging these equations, one obtains the relation

$$\Delta T = \frac{t_{ET=UT} - t_{UT}}{1 - a \cdot 0^\circ 25/m} \quad , \quad (10.8)$$

from which ΔT may be estimated.

Table 10.2. Examples for the determination of the difference between Universal Time and Ephemeris Time from stellar occultations

Stern	Observing site λ φ	Observing time		Computed ET=UT	a	ΔT
		Date	UT			
17 Tau	Roy. Obs. Berlin -13°396 +52°505	26.09.1896	20:32:10	20:32:06	+0 ^m 7/°	-5 s
57B Sco	Univ. Warsaw -21°030 +52°218	24.01.1930	05:30:45	05:31:10	+0 ^m 3/°	27 s
μ Cet	Obs. Stuttgart -9°197 +48°784	10.02.1962	21:12:45	21:13:21	-0 ^m 6/°	32 s
δ Psc	Obs. Stuttgart -9°197 +48°784	13.01.1970	19:35:12	19:36:02	-1 ^m 1/°	39 s
ϵ Ari	St. Augustin -7°177 +50°775	30.11.1990	22:01:32	22:03:01	-1 ^m 9/°	60 s

As an example, five stellar occultations made between 1896 and 1990 that have been observed at various European sites are listed in Table 10.2. It is easy to see that the difference between Ephemeris Time and Universal Time has increased by approximately one minute over the last century, which is a visible sign of the deceleration of the Earth's rotation. Compared to the actual values, the ΔT values estimated from the observations exhibit differences of 2–3 seconds. These errors correspond to the accuracy of predictions made by OCCULT, and are essentially caused by small errors in the stellar and lunar coordinates and the neglect of the lunar limb profile.

11. Orbit Determination

The classical aim of orbit determination is to obtain the orbital elements of a planet, comet, or minor planet from the smallest possible number of observed positions. This is therefore essentially the opposite to determining an ephemeris, where positions are obtained from known orbital elements. Any observation made from Earth at a specific time gives two spherical coordinates. We may choose whether these relate to the celestial equator (right ascension and declination) or to the ecliptic (ecliptic longitude and latitude). On the other hand, the distance cannot be measured, so knowledge of it cannot be used in orbit determination. To derive six orbital elements, the same number of independent observational values are required, so three observations must be available.

Bucerius' method of orbit determination described here, is mainly derived from that of Gauss, but certain points have been considerably simplified. There are some restrictions on its use, therefore, but it is easier to understand and to employ.

11.1 Determining an Orbit from Two Position Vectors

The orbital elements are generally employed to describe a planetary or cometary orbit, because they enable one to obtain a particularly clear interpretation of the individual values. For orbit determination, however, it is more convenient to use another method of description. The orbit is equally well defined if we know the position and velocity of a celestial body at a specific instant, or alternatively, two positions on the orbit and the times at which they were reached. The former description is used in Laplace's method of orbit determination, but Gauss' method employs two position vectors. This section will deal with calculating the elements of an orbit that is described by two known positions $\mathbf{r}_a$ and $\mathbf{r}_b$ at times t_a and t_b . The problem of determining an orbit then reduces to determining two heliocentric positions from three observed directions.

First, we will discuss the intermediate step of calculating the *sector-triangle ratio*, which is the most difficult part of determining an orbit. Yet this value plays an exceedingly important part in the later steps in determining an orbit, as will be seen from later sections.

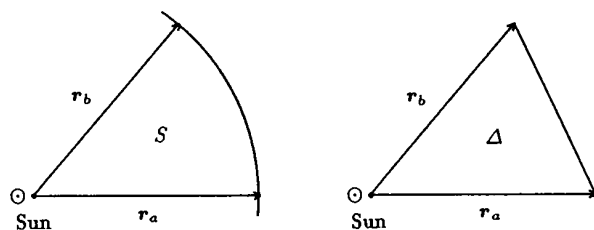


Fig. 11.1. Areas of sectors and triangles

11.1.1 The Sector-Triangle Ratio

The area Δ of the triangle defined by the vectors $\mathbf{r}_a$ and $\mathbf{r}_b$ (Fig. 11.1) depends on the length of the sides r_a and r_b , and the included angle $\nu_b - \nu_a$, which must, however, always be less than 180° in what follows:

$$\Delta = \frac{1}{2} r_a r_b \cdot \sin(\nu_b - \nu_a) \quad . \quad (11.1)$$

Here ν_a and ν_b are the values of the true anomaly at the end points of the portion of the orbit under consideration.

The area S of the sector that is bounded by $\mathbf{r}_a$ and $\mathbf{r}_b$ and the arc of the orbit between them, is – because of Kepler's Second Law (the law of areas) – proportional to the difference between the times t_a and t_b :

$$S = \frac{1}{2} \sqrt{GM_\odot} \cdot \sqrt{a(1-e^2)} \cdot (t_b - t_a) \quad . \quad (11.2)$$

Here a and e denote the semi-major axis and the eccentricity of the orbit that joins the given points (see Chap. 4). If we substitute the semi-latus rectum $p = a(1-e^2)$, then we obtain the expression

$$\eta = \frac{S}{\Delta} = \frac{\sqrt{p} \cdot \tau}{r_a r_b \cdot \sin(\nu_b - \nu_a)} \quad , \quad (11.3)$$

for the ratio η between the two areas, where, for simplicity, the interval is defined by

$$\tau = \sqrt{GM_\odot} \cdot (t_b - t_a) \quad . \quad (11.4)$$

As will be seen, the equation for η contains the semi-latus rectum p , which has not previously been expressed in terms of $\mathbf{r}_a$ and $\mathbf{r}_b$. If we try to eliminate the semi-latus rectum by using the known equations for the two-body problem, then we find that it is no longer possible to express η as a solvable algebraic equation. Instead we obtain the transcendental equation¹

$$\eta = 1 + \frac{m}{\eta^2} \cdot W \left(\frac{m}{\eta^2} - l \right) \quad , \quad (11.5)$$

¹A derivation of this equation would occupy several pages and is therefore omitted here. The reader will find the details in one of the reference works included in the bibliography.

with the (positive) auxiliary variables

$$\begin{aligned} m &= \frac{\tau^2}{\sqrt{2(r_a r_b + \mathbf{r}_a \cdot \mathbf{r}_b)^3}} \\ l &= \frac{r_a + r_b}{2\sqrt{2(r_a r_b + \mathbf{r}_a \cdot \mathbf{r}_b)}} - \frac{1}{2}, \end{aligned} \quad (11.6)$$

from which η has to be determined. Here the function W is defined by:

$$W(w) = \begin{cases} \frac{2g - \sin(2g)}{\sin^3(g)}, & g = 2 \arcsin \sqrt{w} & 0 < w < 1 \\ \frac{4}{3} + \frac{4 \cdot 6}{3 \cdot 5} w + \frac{4 \cdot 6 \cdot 8}{3 \cdot 5 \cdot 7} w^2 + \dots & w \approx 0 \\ \frac{\sinh(2g) - 2g}{\sinh^3(g)}, & g = 2 \operatorname{arsinh} \sqrt{-w} & w < 0 \end{cases} \quad (11.7)$$

To determine η iteratively, it is appropriate to use the secant procedure. If we write

$$f(x) = 1 - x + \frac{m}{x^2} \cdot W\left(\frac{m}{x^2} - l\right),$$

then the desired value of η is that at which the function f becomes zero. Using two approximations η_{i-1} and η_i , we obtain an improved value η_{i+1} via

$$\eta_{i+1} = \eta_i - f(\eta_i) \cdot \frac{\eta_i - \eta_{i-1}}{f(\eta_i) - f(\eta_{i-1})}.$$

Geometrically, this gives us the zero-point of the secant that passes through the points $(\eta_{i-1}, f(\eta_{i-1}))$ and $(\eta_i, f(\eta_i))$ on the curve given by f . If this step is repeatedly carried out, the iteration soon converges to the desired value of the sector-triangle ratio. Suitable starting values

$$\eta_1 = \eta_{\text{Hansen}} + 0.1 \quad \text{and} \quad \eta_2 = \eta_{\text{Hansen}}$$

are given by what is known as the Hansen Approximation

$$\eta_{\text{Hansen}} = \frac{12}{22} + \frac{10}{22} \sqrt{1 + \frac{44}{9} \frac{m}{l + 5/6}}. \quad (11.8)$$

In order to program the equations given above, the data type

```
TYPE INDEX = (X,Y,Z);
VECTOR = ARRAY[INDEX] OF REAL;
```

and the functions

```

(*-----*)
(* DOT: dot product of two vectors *)
(*-----*)
FUNCTION DOT(A,B:VECTOR):REAL;
  BEGIN
    DOT := A[X]*B[X]+A[Y]*B[Y]+A[Z]*B[Z];
  END;
(*-----*)
(* NORM: magnitude of a vector *)
(*-----*)
FUNCTION NORM(A:VECTOR):REAL;
  BEGIN
    NORM := SQRT(DOT(A,A));
  END;
(*-----*)

```

must first be defined. We then obtain the following sub-routine for calculating the sector-triangle ratio.

```

(*-----*)
(* FIND_ETA: determines the sector/triangle ratio *)
(* from two positions and the time difference *)
(*-----*)
FUNCTION FIND_ETA ( RA,RB: VECTOR; TAU: REAL ): REAL;

  CONST DELTA = 1.0E-9;  MAXIT = 30;

  VAR  KAPPA,M,L,SA,SB,ETA_MIN,ETA1,ETA2,F1,F2,D_ETA: REAL;
       I: INTEGER;

  (* F(eta) = 1 - eta + (m/eta**2)*W(m/eta**2-1) *)
  FUNCTION F ( ETA,M,L: REAL ): REAL;
    CONST EPS = 1.0E-10;
    VAR W,WW,A,S,N,G,E: REAL;
    BEGIN
      W := M/(ETA*ETA)-L;
      IF (ABS(W)<0.1)
      THEN (* series expansion *)
        BEGIN
          A:=4.0/3.0; WW:=A; N:=0.0;
          REPEAT
            N:=N+1; A:=A*W*(N+2.0)/(N+1.5); WW:=WW+A;
          UNTIL ABS(A)<EPS;
        END
      ELSE
        IF (W>0)
        THEN (* W=(2g-sin2g)/(sin(g)**3), g=2*arcsin(sqrt(w)) *)
          BEGIN
            G := 2.0*ARCTAN(SQRT(W/(1.0-W))); S := SIN(G);
            WW := (2.0*G-SIN(2.0*G))/(S*S*S);
          END
        ELSE (* W=(sinh2g-2g)/(sinh(g)**3), g=2*arsinh(sqrt(-w)) *)
          BEGIN

```

```

      G := 2.0*LN(SQRT(-W)+SQRT(1.0-W));
      E := EXP(G); S:=0.5*(E-1.0/E); E:=E*E;
      WW := (0.5*(E-1.0/E)-2.0*G)/(S*S*S);
      END;
      F := 1.0-ETA+(W+L)*WW;
      END; (* FIND_ETA.F *)

BEGIN

  SA := NORM(RA); SB := NORM(RB); KAPPA := SQRT(2.0*(SA*SB+DOT(RA,RB)));
  M := TAU*TAU / (KAPPA*KAPPA*KAPPA); L := (SA+SB)/(2.0*KAPPA) - 0.5;
  ETA_MIN := SQRT(M/(L+1.0));

  (* start with Hansen's approximation *)
  ETA2 := ( 12.0 + 10.0*SQRT(1.0+(44.0/9.0)*M/(L+5.0/6.0)) ) / 22.0;
  ETA1 := ETA2 + 0.1; F1 := F(ETA1,M,L); F2 := F(ETA2,M,L); I := 0;

  (* secant method *)
  WHILE ( ABS(F2-F1)>DELTA) AND (I<MAXIT) ) DO
    BEGIN
      D_ETA:=-F2*(ETA2-ETA1)/(F2-F1); ETA1:=ETA2; F1:=F2;
      WHILE (ETA2+D_ETA<=ETA_MIN) DO D_ETA:=0.5*D_ETA;
      ETA2:=ETA2+D_ETA; F2:=F(ETA2,M,L); I:=I+1;
    END;
  IF (I=MAXIT) THEN WRITELN(' convergence problems in FIND_ETA');
  FIND_ETA := ETA2;

END;

```

(*-----*)

Because Pascal does not recognize hyperbolic functions, these are expressed in terms of exponential and logarithmic functions:

$$\begin{aligned}\sinh x &= \frac{1}{2}(\exp x - 1/\exp x) \\ \sinh(2x) &= \frac{1}{2}((\exp x)^2 - 1/(\exp x)^2) \\ \operatorname{arsinh} x &= \ln(x + \sqrt{1+x^2}) \quad .\end{aligned}$$

In a similar way, we also have

$$\arcsin(x) = \arctan \frac{x}{\sqrt{1-x^2}} \quad .$$

11.1.2 Orbital Elements

The orbit of a celestial body that passes through the points $\mathbf{r}_a$ and $\mathbf{r}_b$, is always restricted to the plane determined by these two points and the Sun. In order to determine the inclination i of this plane to the ecliptic as well as

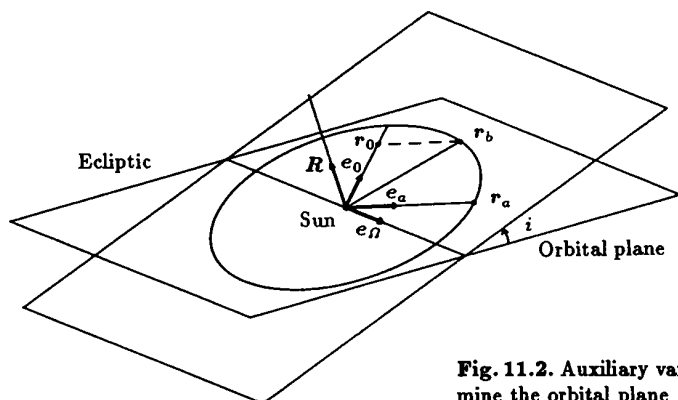


Fig. 11.2. Auxiliary variables used to determine the orbital plane

the longitude of the ascending node, we first obtain the unit vectors e_a and e_0 , which both lie in the orbital plane:

$$e_a = \frac{\mathbf{r}_a}{|\mathbf{r}_a|} \quad (11.9)$$

$$e_0 = \frac{\mathbf{r}_0}{|\mathbf{r}_0|} \quad \text{where} \quad \mathbf{r}_0 = \mathbf{r}_b - (\mathbf{r}_b \cdot e_a)e_a \quad (11.10)$$

The nature of these vectors is shown in Fig. 11.2. e_a is aligned with $\mathbf{r}_a$, $\mathbf{r}_0$ and e_0 are perpendicular to it. If we now form the cross-product of e_a and e_0 , the result obtained is the Gaussian vector $\mathbf{R}$, which is perpendicular to the orbital plane and is likewise normalized to unit length ($|\mathbf{R}| = 1$):

$$\mathbf{R} = e_a \times e_0, \quad \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} y_a z_0 - z_a y_0 \\ z_a x_0 - x_a z_0 \\ x_a y_0 - y_a x_0 \end{pmatrix} \quad (11.11)$$

$\mathbf{R}$ is directed to ecliptic longitude $l = \Omega - 90^\circ$ and ecliptic latitude $b = 90^\circ - i$, and can therefore be expressed in terms of the elements Ω and i :

$$\mathbf{R} = \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} +\cos(90^\circ - i) \cos(\Omega - 90^\circ) \\ +\cos(90^\circ - i) \sin(\Omega - 90^\circ) \\ +\sin(90^\circ - i) \end{pmatrix} = \begin{pmatrix} +\sin i \sin \Omega \\ -\sin i \cos \Omega \\ +\cos i \end{pmatrix} \quad (11.12)$$

We thus obtain three equations, which may be solved unambiguously for the longitude of the node and the orbital inclination:

$$\Omega = 90^\circ + \arctan(R_y/R_x) = \arctan(-R_x/R_y) \quad (11.13)$$

$$i = 90^\circ - \arcsin(R_z) \quad (11.14)$$

Both angles refer to the same equinox as the vectors $\mathbf{r}_a$ and $\mathbf{r}_b$. From the position of the line of nodes the argument of latitude u_a may now be determined, it being given by the angle between the position vector $\mathbf{r}_a$ and the

direction of the ascending node of the orbit. For this angle we have

$$\begin{aligned}\cos u_a &= \mathbf{e}_a \cdot \mathbf{e}_\Omega = x_a \cdot \cos \Omega + y_a \cdot \sin \Omega \\ \cos(u_a + 90^\circ) &= \mathbf{e}_0 \cdot \mathbf{e}_\Omega = x_0 \cdot \cos \Omega + y_0 \cdot \sin \Omega \quad ,\end{aligned}$$

where

$$\mathbf{e}_\Omega = \begin{pmatrix} \cos \Omega \\ \sin \Omega \\ 0 \end{pmatrix}$$

is the unit vector directed along the line of nodes. We therefore have

$$\begin{aligned}u_a &= \arctan \left(\frac{-x_0 \cdot \cos \Omega - y_0 \cdot \sin \Omega}{+x_a \cdot \cos \Omega + y_a \cdot \sin \Omega} \right) \\ &= \arctan \left(\frac{+x_0 \cdot R_y - y_0 \cdot R_x}{-x_a \cdot R_y + y_a \cdot R_x} \right) \quad .\end{aligned} \quad (11.15)$$

To determine the remaining orbital elements we require the sector-triangle ratio, which is calculated in the manner described in the previous section. Using this, we are next able to express the semi-latus rectum

$$p = \left(\frac{2 \cdot \Delta \cdot \eta}{\tau} \right)^2$$

in terms of the area of the triangle defined by the vectors $\mathbf{r}_a$ and $\mathbf{r}_b$

$$\Delta = \frac{1}{2} r_a r_b \cdot \sin(\nu_b - \nu_a) = \frac{1}{2} r_a r_0 \quad ,$$

and the interval τ .

The shape of the orbit is defined by the eccentricity e , which may be determined from the equation for the conic section

$$r = \frac{p}{1 + e \cdot \cos \nu} \quad .$$

Solving for $e \cos \nu$, we have

$$\begin{aligned}e \cdot \cos \nu_a &= p/r_a - 1 \\ e \cdot \cos \nu_b &= p/r_b - 1 \quad .\end{aligned}$$

If we take into account the fact that

$$\begin{aligned}\cos \nu_b &= \cos \nu_a \cos(\nu_b - \nu_a) - \sin \nu_a \sin(\nu_b - \nu_a) \\ &= \cos \nu_a \cdot \left(\frac{\mathbf{r}_b \cdot \mathbf{e}_a}{r_b} \right) - \sin \nu_a \cdot \left(\frac{r_0}{r_b} \right) \quad ,\end{aligned}$$

then, by substitution, we obtain the two equations

$$\begin{aligned}e \cdot \cos \nu_a &= p/r_a - 1 \\ e \cdot \sin \nu_a &= \left\{ (p/r_a - 1) \left(\frac{\mathbf{r}_b \cdot \mathbf{e}_a}{r_b} \right) - (p/r_b - 1) \right\} / \left(\frac{r_0}{r_b} \right) \quad ,\end{aligned}$$

which may themselves be solved for the eccentricity and the true anomaly at time t_a :

$$e = \sqrt{(e \cdot \cos(\nu_a))^2 + (e \cdot \sin(\nu_a))^2}$$

$$\nu_a = \arctan\left(\frac{e \cdot \sin(\nu_a)}{e \cdot \cos(\nu_a)}\right) .$$

The argument and longitude of perihelion are obtained from the difference between the argument of latitude and the true anomaly, being

$$\omega = u_a - \nu_a \quad (11.16)$$

$$\varpi = u_a - \nu_a + \Omega . \quad (11.17)$$

The eccentricity enables us to determine whether the orbit between $\mathbf{r}_a$ and $\mathbf{r}_b$ is an ellipse ($e < 1$) or a hyperbola ($e > 1$). Parabolic orbits are not considered further here, because in determining an orbit in practice it is highly unlikely that e will turn out to be exactly equal to one. From the semi-latus rectum and the eccentricity we can now also obtain the semi-major axis and the perihelion distance:

$$a = \frac{p}{1 - e^2} \quad (11.18)$$

$$q = \frac{p}{1 + e} . \quad (11.19)$$

Here it should be noted that, by definition, the semi-major axis of a hyperbolic orbit is a negative value.

We now know all the orbital elements that determine the orbit's shape (e), size (a), and orientation in space (i, Ω, ω). The sixth and last element to be determined is therefore the time of perihelion passage, which defines when the body will pass (or passed) the point closest to the Sun. For elliptical orbits, we first need to determine the eccentric anomaly E_a from the equations

$$\cos E_a = \frac{\cos \nu_a + e}{1 + e \cdot \cos \nu_a}$$

$$\sin E_a = \frac{\sqrt{1 - e^2} \sin \nu_a}{1 + e \cdot \cos \nu_a} .$$

We obtain these equations by eliminating the radius from (4.5) by using the equation for the conic section, and then solving for the eccentric anomaly. The value for the mean anomaly M corresponding to E is obtained from the Kepler equation

$$M_a = E_a - e \cdot \sin E_a \quad (\text{radians}) .$$

For an orbital period of

$$T = 2\pi \cdot \sqrt{\frac{a^3}{GM_\odot}} ,$$

as obtained from Kepler's Third Law, the mean anomaly varies daily by

$$n = 2\pi \cdot \frac{1^d}{T} = \sqrt{\frac{GM_\odot}{a^3}} \cdot 1^d ,$$

so the time of perihelion passage is given by

$$t_0 = t_a - M_a / \sqrt{GM_\odot / a^3} . \quad (11.20)$$

For hyperbolic orbits the corresponding equations are

$$\sinh H_a = \frac{\sqrt{e^2 - 1} \sin \nu_a}{1 + e \cdot \cos \nu_a} \quad (11.21)$$

$$M_a = e \cdot \sinh H_a - H_a \quad (11.22)$$

$$t_0 = t_a - M_a / \sqrt{GM_\odot / |a|^3} . \quad (11.23)$$

The complete method of determining orbital elements from two given points on the orbit is given in the procedure **ELEMENT**. In order to facilitate data entry, $\mathbf{r}_a$ and $\mathbf{r}_b$ are again combined as fields of type **VECTOR**, which was defined earlier.

```
(*-----*)
(* CROSS: cross product of two vectors *)
(*-----*)
```

```
PROCEDURE CROSS(A,B:VECTOR;VAR C:VECTOR);
```

```
  BEGIN
```

```
    C[X] := A[Y]*B[Z]-A[Z]*B[Y];
```

```
    C[Y] := A[Z]*B[X]-A[X]*B[Z];
```

```
    C[Z] := A[X]*B[Y]-A[Y]*B[X];
```

```
  END;
```

```
(*-----*)
(* ELEMENT: calculates orbital elements from two positions *)
(*      for elliptic and hyperbolic orbits *)
(*-----*)
(* JDA,JDB: time of passage of points A and B (Julian Date) *)
(* RA, RB : position vectors of points A and B *)
(* TP : perihelion time (in Julian centuries since J2000) *)
(* Q : perihelion distance *)
(* ECC : eccentricity *)
(* INC : inclination (in deg) *)
(* LAN : longitude of the ascending node (in deg) *)
(* AOP : argument of perihelion (in deg) *)
(*-----*)
```

```
PROCEDURE ELEMENT ( JDA,JDB: REAL; RA,RB: VECTOR;
                    VAR TP,Q,ECC,INC,LAN,AOP: REAL);
```

```
  CONST KGAUSS = 0.01720209895;
```

```
    RAD = 0.01745329252; (* 180/pi *)
```

```
  VAR TAU,ETA,P,AX,N,NY,E,M,U : REAL;
```

```

SA,SB,SO,FAC,DUMMY,SHH      : REAL;
COS_DNY,SIN_DNY,ECOS_NY,ESIN_NY : REAL;
EA,RO,EO,R                  : VECTOR;
I                             : INDEX;

```

```

BEGIN

```

```

(* calculate vector RO (fraction of RB perpendicular to RA) *)
(* and the magnitudes of RA, RB and RO *)

```

```

SA := NORM(RA);   FOR I:=X TO Z DO  EA[I]:=RA[I]/SA;
SB := NORM(RB);
FAC := DOT(RB,EA); FOR I:=X TO Z DO  RO[I]:=RB[I]-FAC*EA[I];
SO := NORM(RO);   FOR I:=X TO Z DO  EO[I]:=RO[I]/SO;

```

```

(* inclination and ascending node *)

```

```

CROSS (EA,EO,R);
POLAR ( -R[Y],R[X],R[Z], DUMMY,INC,LAN );  INC := 90.0-INC;
U := ATN2 ( (+EO[X]*R[Y]-EO[Y]*R[X]) , (-EA[X]*R[Y]+EA[Y]*R[X]) );
IF INC=0.0 THEN U:=ATN2(RA[Y],RA[X]);

```

```

(* semilatus rectum p *)

```

```

TAU := KGAUSS * ABS(JDB-JDA);  ETA := FIND_ETA(RA,RB,TAU);
P := SA*SO*ETA / TAU;  P := P*P;

```

```

(* eccentricity, true anomaly and longitude of perihelion *)

```

```

COS_DNY := FAC/SB;  SIN_DNY := SO/SB;
ECOS_NY := P/SA-1.0;  ESIN_NY := (ECOS_NY+COS_DNY-(P/SB-1.0))/SIN_DNY;
POLAR ( ECOS_NY,ESIN_NY,0.0, ECC,DUMMY,NY );
AOP := U-NY;  WHILE (AOP<0.0) DO AOP:=AOP+360.0;

```

```

(* perihelion distance, semimajor axis and mean daily motion *)

```

```

Q := P/(1.0+ECC);  AX := Q/(1.0-ECC);
N := KGAUSS / SQRT(ABS(AX*AX*AX));

```

```

(* mean anomaly and time of perihelion passage *)

```

```

IF (ECC<1.0)

```

```

  THEN

```

```

    BEGIN

```

```

      E := ATN2 ( SQRT((1.0-ECC)*(1.0+ECC))*ESIN_NY, ECOS_NY+ECC*ECC );
      E := RAD*E;  M := E-ECC*SIN(E); ;

```

```

    END

```

```

  ELSE

```

```

    BEGIN

```

```

      SHH := SQRT((ECC-1.0)*(ECC+1.0))*ESIN_NY / (ECC+ECC*ECOS_NY) ;
      M := ECC*SHH - LN(SHH+SQRT(1.0+SHH*SHH))

```

```

    END;

```

```

TP := ( (JDA-M/N) - 2451545.0 ) / 36525.0;

```

```

END;

```

```

(*-----*)

```


11.2 The Geometry of Geocentric Observations

If the vector $\mathbf{r}$ is the position of a planet relative to the Sun, and $\mathbf{R}$ the geocentric position of the Sun, then the planet's geocentric position is given by

$$\rho \mathbf{e} = \mathbf{R} + \mathbf{r} \quad . \quad (11.24)$$

Here ρ is the distance of the planet from the Earth, and $\mathbf{e}$ is a vector of unit length, directed from the Earth towards the planet. In ecliptic or equatorial coordinates, $\mathbf{e}$ has the components

$$\begin{pmatrix} \cos \lambda \cos \beta \\ \sin \lambda \cos \beta \\ \sin \beta \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{pmatrix} \quad ,$$

where λ and β are the ecliptic longitude and latitude and α and δ are the right ascension and declination of the planet as seen from Earth.

If the coordinates of the planet on the apparent celestial sphere (i.e., α and δ) are determined, then the observing direction – and thus $\mathbf{e}$ – is defined. The distance ρ is unknown, and must be calculated during the process of determining the orbit. For a planetary orbit to be determined unambiguously at least three observations $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ must be available. We may also assume that the coordinates of the Sun at the times of observation $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$, are known. We must try to calculate the distances ρ_1 , ρ_2 and ρ_3 from these data. Only when the distances are known can we determine the heliocentric positions that define the orbit and derive the orbital elements. The geometrical relationships that are obtained from a set of three planetary positions, now need to be expressed in a form suitable for orbit determination.

At the times $t_1 < t_2 < t_3$ the planet is at positions $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ with respect to the Sun. Since for unperturbed Keplerian motion all the points and the Sun lie in a single plane, it is always possible to express one position vector using an appropriate combination of the other two. For this purpose we choose $\mathbf{r}_2$, and may then write

$$\mathbf{r}_2 = n_1 \mathbf{r}_1 + n_3 \mathbf{r}_3 \quad (\text{equation of the orbital plane}) \quad . \quad (11.25)$$

The factors n_1 and n_3 depend on the relative position of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$. In what follows, if we assume that the entire arc of the orbit is less than 180° , then both factors are positive. We now combine (11.24) and (11.25), obtaining

$$(\rho_2 \mathbf{e}_2 - \mathbf{R}_2) = n_1 \cdot (\rho_1 \mathbf{e}_1 - \mathbf{R}_1) + n_3 \cdot (\rho_3 \mathbf{e}_3 - \mathbf{R}_3)$$

or, rearranging,

$$n_1 \rho_1 \mathbf{e}_1 - \rho_2 \mathbf{e}_2 + n_3 \rho_3 \mathbf{e}_3 = n_1 \mathbf{R}_1 - \mathbf{R}_2 + n_3 \mathbf{R}_3 \quad . \quad (11.26)$$

If we now define the vectors

$$\mathbf{d}_1 = \mathbf{e}_2 \times \mathbf{e}_3 \quad \mathbf{d}_2 = \mathbf{e}_3 \times \mathbf{e}_1 \quad \mathbf{d}_3 = \mathbf{e}_1 \times \mathbf{e}_2 \quad ,$$

then, because of the properties of the cross-products, $\mathbf{d}_1$ is perpendicular to $\mathbf{e}_2$ and $\mathbf{e}_3$, $\mathbf{d}_2$ is perpendicular to $\mathbf{e}_3$, and $\mathbf{e}_1$ and $\mathbf{d}_3$ perpendicular to $\mathbf{e}_1$ and $\mathbf{e}_2$. Consequently, the dot product $\mathbf{e}_i \cdot \mathbf{d}_j$ only differs from zero for $i = j$. If we multiply (11.26) by $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3$ individually, then we obtain the equations

$$\begin{aligned} n_1 \rho_1 \cdot (\mathbf{e}_1 \cdot \mathbf{d}_1) &= (n_1 \mathbf{R}_1 - \mathbf{R}_2 + n_3 \mathbf{R}_3) \cdot \mathbf{d}_1 \\ -\rho_2 \cdot (\mathbf{e}_2 \cdot \mathbf{d}_2) &= (n_1 \mathbf{R}_1 - \mathbf{R}_2 + n_3 \mathbf{R}_3) \cdot \mathbf{d}_2 \\ n_3 \rho_3 \cdot (\mathbf{e}_3 \cdot \mathbf{d}_3) &= (n_1 \mathbf{R}_1 - \mathbf{R}_2 + n_3 \mathbf{R}_3) \cdot \mathbf{d}_3 \quad . \end{aligned}$$

If we introduce the abbreviations

$$\begin{aligned} D &= \mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3) = \mathbf{e}_2 \cdot (\mathbf{e}_3 \times \mathbf{e}_1) = \mathbf{e}_3 \cdot (\mathbf{e}_1 \times \mathbf{e}_2) \\ &= \mathbf{e}_1 \cdot \mathbf{d}_1 = \mathbf{e}_2 \cdot \mathbf{d}_2 = \mathbf{e}_3 \cdot \mathbf{d}_3 \end{aligned}$$

and

$$D_{ij} = \mathbf{d}_i \cdot \mathbf{R}_j \quad ,$$

then we obtain the three equations

$$\begin{aligned} \rho_1 &= \frac{1}{n_1 D} (n_1 D_{11} - D_{12} + n_3 D_{13}) \\ \rho_2 &= \frac{1}{-D} (n_1 D_{21} - D_{22} + n_3 D_{23}) \\ \rho_3 &= \frac{1}{n_3 D} (n_1 D_{31} - D_{32} + n_3 D_{33}) \quad . \end{aligned} \tag{11.27}$$

The distances ρ_1 , ρ_2 and ρ_3 can therefore be expressed in terms of n_1 and n_3 as well as the vectors $\mathbf{e}_{1,3}$ and $\mathbf{R}_{1,3}$. At first it would seem that we have not gained much, because n_1 and n_3 are unknown. By using the equation of the orbital plane, we have at least reduced the number of unknowns from three ($\rho_{1,3}$) to two ($n_{1,3}$). The newly introduced coefficients are particularly important, however, because – as will now be shown – they can be closely approximated by expressions involving the known intervals between the observations.

Consider the equation of the orbital plane (11.25). If we form the cross-product of both sides with $\mathbf{r}_3$ or $\mathbf{r}_1$, and, remind ourselves that the cross-product of a vector with itself cancels out, we obtain the expressions

$$(\mathbf{r}_2 \times \mathbf{r}_3) = n_1 \cdot (\mathbf{r}_1 \times \mathbf{r}_3) \quad (\mathbf{r}_1 \times \mathbf{r}_2) = n_3 \cdot (\mathbf{r}_1 \times \mathbf{r}_3)$$

and

$$n_1 = \frac{|\mathbf{r}_2 \times \mathbf{r}_3|}{|\mathbf{r}_1 \times \mathbf{r}_3|} \quad n_3 = \frac{|\mathbf{r}_1 \times \mathbf{r}_2|}{|\mathbf{r}_1 \times \mathbf{r}_3|} \quad .$$

Generalizing, as the area Δ of the triangle bounded by two vectors $\mathbf{r}_a$ and $\mathbf{r}_b$, is equal to

$$\Delta = \frac{1}{2} |\mathbf{r}_a \times \mathbf{r}_b| \quad ,$$

n_1 and n_3 can be interpreted as expressions of the areas of the triangles bounded by $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ (see Fig. 11.3):

$$n_1 = \frac{\Delta_1}{\Delta_2} \quad n_3 = \frac{\Delta_3}{\Delta_2} \quad .$$

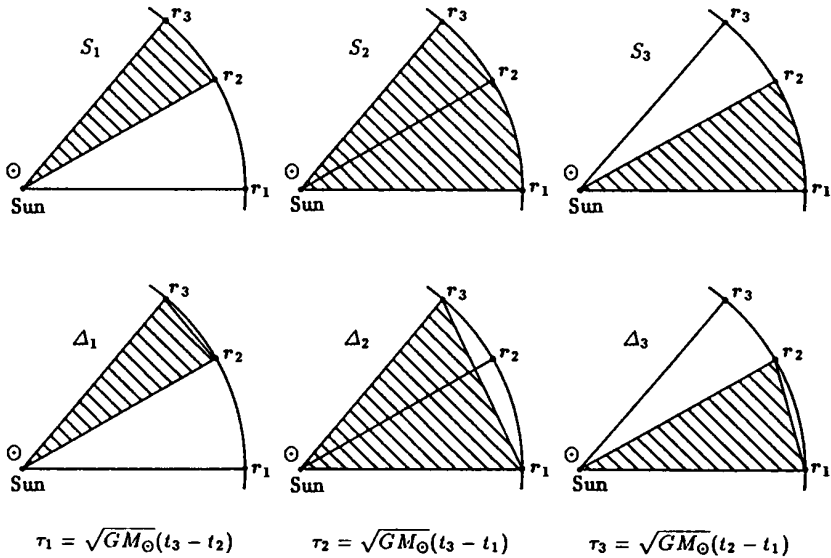


Fig. 11.3. Sector areas, triangle areas, and intervals for three heliocentric positions

For small arcs of the orbit in particular, the areas of the triangles differ only slightly from the corresponding sector areas $S_i = \eta_i \Delta_i$, which are themselves proportional to the intervals τ_i :

$$n_1 = \frac{\eta_2}{\eta_1} \cdot \frac{\tau_1}{\tau_2} \approx \frac{\tau_1}{\tau_2} \quad n_3 = \frac{\eta_2}{\eta_3} \cdot \frac{\tau_3}{\tau_2} \approx \frac{\tau_3}{\tau_2} \quad .$$

We do, therefore, at least know approximate values for n_1 and n_3 , which enable us to determine first approximations for the geocentric distances (ρ_1, ρ_2, ρ_3) .

11.3 The Method of Successive Improvement

Two statements essentially summarize the various considerations that we have covered so far: If we know the heliocentric position of a celestial body at

two given times, then the whole orbit is unambiguously determined. The same applies to the orbital elements and the sector-triangle ratio. If, on the other hand, we know the value of the sector-triangle ratio for a set of three observed positions, then the geocentric and heliocentric position vectors may be calculated. The shortened Gaussian method of orbit determination to be described now is based upon this.

Let three geocentric observed directions (e_1, e_2, e_3), the corresponding geocentric coordinates of the Sun (R_1, R_2, R_3) and the intervals (τ_1, τ_2, τ_3) be given. The steps in the calculation, using these initial data, are:

1. Set $n_1 = \tau_1/\tau_2$ and $n_3 = \tau_3/\tau_2$ as initial approximations for the ratios of the areas of the triangles.
2. Repeat steps (a)...(d), until n_1, n_3 and the other values no longer vary significantly.
 - (a) Calculate the geocentric distances (ρ_1, ρ_2, ρ_3) with (11.27).
 - (b) Thence calculate the heliocentric position vectors (r_1, r_2, r_3) from (11.24).
 - (c) Calculate the sector-triangle ratios (η_1, η_2, η_3) from each pair of heliocentric position vectors and the corresponding interval with (11.5).
 - (d) Calculate improved values $n_1 = (\eta_2/\eta_1) \cdot (\tau_1/\tau_2)$ and $n_3 = (\eta_2/\eta_3) \cdot (\tau_3/\tau_2)$ for the triangle area ratios.
3. Calculate the orbital elements from the last values of r_1, r_3 and τ_2 .

The orbit calculated in this way has the properties desired, in that any body moving along it is located at the times t_1, t_2 and t_3 at the points r_1, r_2 and r_3 , which themselves lie in the directions e_1, e_2 and e_3 relative to the Earth.

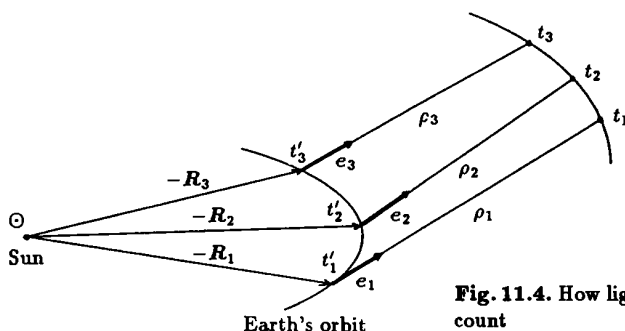


Fig. 11.4. How light-time is taken into account

We have therefore, in principle, solved the problem of determining an orbit. Only one small change is required for this method to be employed in practice. The way in which the problem of determining an orbit has been

handled until now has assumed that the observed position of a planet or comet is always identical with the instantaneous, geometrical position. In reality, however, there is a certain lapse of time between the light being emitted and its being observed, and this interval is determined by the finite speed of light

$$c = 173.1 \text{ AU/d} \\ 1/c = 0^d05776/\text{AU} \quad .$$

The vectors $\mathbf{e}_i$, which represent the directions of observation, therefore are directed from the heliocentric positions $-\mathbf{R}_i$, where the Earth is located at the times of observation, to the three points $\mathbf{r}_i = \rho_i \mathbf{e}_i - \mathbf{R}_i$, where the planet was at the slightly earlier times $t_i < t'_i$ (see Fig. 11.4). The time $t'_i - t_i$ that elapses between when the light is emitted and when it is observed, is thus equal to the time light requires to cover the distance ρ_i :

$$\rho_i = c \cdot (t'_i - t_i) \quad .$$

We immediately have the problem that the times of emission of the light t_i that are required for calculating the intervals τ_i are unknown. In the course of determining the orbit, however, we obtain increasingly accurate values for the geocentric distances (ρ_1, ρ_2, ρ_3) , so the effect of light-time may be regarded as not posing any problems. We employ the observational times solely in determining the initial approximations

$$n_1 = \frac{\tau_1}{\tau_2} \approx \frac{t'_3 - t'_2}{t'_3 - t'_1} \quad n_3 = \frac{\tau_3}{\tau_2} \approx \frac{t'_2 - t'_1}{t'_3 - t'_1}$$

for the ratios of the areas of the triangles. Afterwards the intervals are recalculated at each iteration, as soon as the actual geocentric distances are known:

$$\tau_1 = \sqrt{GM_\odot}(t_3 - t_2) \quad \tau_2 = \sqrt{GM_\odot}(t_3 - t_1) \quad \tau_3 = \sqrt{GM_\odot}(t_2 - t_1)$$

using

$$t_i = t'_i - \rho_i \cdot 0^d05776/\text{AU} \quad (i = 1 \dots 3) \quad .$$

The auxiliary geometrical variables $\mathbf{d}_i$, D_i and D depend only on the direction vectors $\mathbf{e}_i$ and the solar coordinates $\mathbf{R}_i$, and therefore do not require correction.

11.4 Multiple Solutions

Apart from exceptional cases when the iteration does not converge, the simplified Gaussian method always provides a result that correctly satisfies all the equations involved in determining an orbit. In doubtful cases it can always be confirmed by appropriate subsequent calculation of the three observations

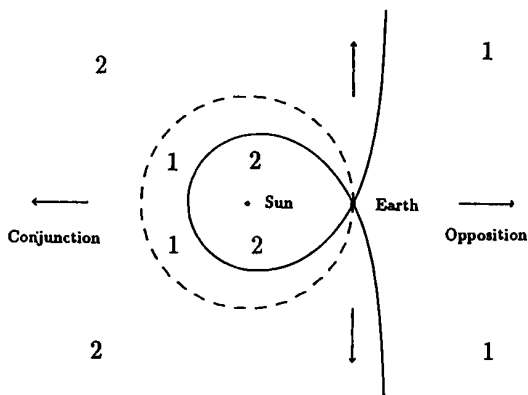


Fig. 11.5. Charlier's boundary line (—) separates regions in which orbit determination is unambiguous (1), from regions in which it is ambiguous (2)

from the orbital elements that have been derived. However, we cannot rule out the fact that at times one obtains solutions that appear, at first sight, false or absurd. This is particularly obvious when we encounter hyperbolic orbits with high eccentricities, which contradict actual experience². The reason for this is that determining orbits from three observations is not always unambiguous. This is particularly easily seen in the *Earth-orbit solution*. The observer is moving in an almost unperturbed Keplerian orbit around the Sun, and is simultaneously always situated in the direction of observation. The quickest way of recognizing the Earth-orbit solution is from the semi-major axis, ($a \approx 1$), the eccentricity ($e \approx 0$), and the inclination to the ecliptic ($i \approx 0^\circ$).

A general discussion of possible ambiguities in orbit determination is normally restricted to small arcs of an orbit. With this assumption, if we consider the relative positions of the Sun, Earth and the observed object, then a total of four separate regions are defined by the Earth's orbit and what is known as *Charlier's boundary line* (Fig. 11.5). For a minor planet that is observed at around the time of opposition, for example, the result is unambiguous, with the exception of the Earth-orbit solution that has been mentioned. If, on the other hand, at the time of observation a comet is within the teardrop-shaped zone that immediately surrounds the Sun, or lies in the region of conjunction outside the Earth's orbit, in principle two solutions to the orbit determination are possible.

The simplified version of the orbit determination procedure that is described here is not, however, capable of calculating all these solutions and discriminating between them. For such a task, reference should be made to the comprehensive Gauss method of orbit determination described in the literature.

²All known comets, without exception, have eccentricities that are smaller than $e = 1.1$.

11.5 The ORBDET Program

Using the ORBDET program described here, the orbital elements of a cometary or minor-planet orbit may be determined from three observations. First the unit vectors of the directions of observation and the coordinates of the Sun are determined from the given values of right ascension and declination (START). As the required orbital elements must, as usual, be relative to the ecliptic, all the values are calculated in a system of ecliptic coordinates, whose equinox may be chosen at liberty. The orbit determination proper, using the simplified Gauss procedure follows, in the GAUSS procedure. Two sub-routines (DUMPELEM, SAVEELEM) write the orbital elements on the screen and to a data file with predefined name ORBOUT(.DAT). The format of this output file corresponds to the format of the data entry in the COMET program in Chap. 4, so it is possible to calculate a subsequent ephemeris without any problems.

Because of the length of the data that have to be entered, these are assembled in an input file ORBINP(.DAT). The data file begins with a single comment line, which enables the data to be easily identified. The three following lines each contain the date (Year, Month, Day, and Hour with decimal fraction), the observed equatorial coordinates (right ascension being given as: Hours, Minutes, and Second with decimal fraction; declination as: Degrees, Minutes, and Seconds with decimal fraction), as well as an optional comment. The next two lines contain the equinox to which the observed positions relate, and the equinox in which the desired orbital elements should be specified. The following data file, for example, contains some observations of Ceres, which Gauss used as an example of his method of determining orbits:

Ceres (Gauss's example)

```
1805 09 05 24.165   6 23 57.54  22 21 27.08  ! Three observations
1806 01 17 22.095   6 45 14.69  30 21 24.20  ! Format: Date (ymdh),
1806 05 23 20.399   8 07 44.60  28 02 47.04  !           RA (hms), Dec (dms)
1806.0              ! Equinox
1806.0              ! Required Equinox
```

In determining orbits all observations are expected to be in the form of astrometric coordinates, i.e., free from the effects of stellar aberration. Amongst others, this is always the case when the coordinates are derived by comparison with surrounding stars from a star catalogue. One example is the measurement of the photograph of a star field, which is the method most frequently employed to follow the movement of a comet or a minor planet.

```
(*-----*)
(*)              ORBDET              (*)
(*)   Gaussian orbit determination from three observations   (*)
(*)           using the abbreviated method of Bucerius       (*)
(*)                   version 93/07/01                        (*)
(*-----*)
```

```
PROGRAM ORBDDET(INPUT,OUTPUT,ORBINP,ORBOU);
```

```
TYPE INDEX = (X,Y,Z);
VECTOR = ARRAY[INDEX] OF REAL;
REAL3 = ARRAY[1.. 3] OF REAL;
REAL33 = ARRAY[1.. 3] OF REAL3;
MAT3X = ARRAY[1.. 3] OF VECTOR;
CHAR80 = ARRAY[1..80] OF CHAR;
VAR TEQX          : REAL;
    TP,Q,ECC,INC,LAN,AOP: REAL;
    JDO          : REAL3;
    RSUN,E       : MAT3X;
    HEADER       : CHAR80;
    ORBINP,ORBOU : TEXT;
```

```
(*-----*)
(* The following procedures should be entered here in the order given *)
(* SN, CS, TN, ATN, ATN2, CART, POLAR, DDD, DMS, DOT, NORM, CROSS *)
(* MJD, CALDAT *)
(* EQUACL, PMATECL, PRECART *)
(* SUN200 *)
(* FIND_ETA, ELEMENT *)
(*-----*)
```

```
(*-----*)
(* START: reads the input file and preprocesses the observational data *)
(* *)
(* output: *)
(* RSUN: matrix of three Sun position vectors in ecliptic coordinates *)
(* E: matrix of three observation direction unit vectors *)
(* JD: julian date of the three observation times *)
(* TEQX: equinox of RSUN and E (in Julian centuries since J2000) *)
(*-----*)
```

```
PROCEDURE START (VAR HEADER: CHAR80;
                 VAR RSUN,E: MAT3X; VAR JDO: REAL3; VAR TEQX: REAL);
```

```
VAR DAY,MONTH,YEAR,D,M,I : INTEGER;
    UT,S,DUMMY          : REAL;
    EQXO,EQX,TEQXO      : REAL;
    LS,BS,RS,LP,BP,RA,DEC,T : REAL3;
    A,AS                : REAL33;
    ORBINP              : TEXT;
```

```
BEGIN
```

```
(* open input file *)

(* RESET(ORBINP); *) (* Standard Pascal *)
ASSIGN(ORBINP,'ORBINP.DAT'); RESET(ORBINP); (* Turbo Pascal *)
(* RESET(ORBINP,'ORBINP.DAT'); *) (* ST Pascal plus *)

(* read data from file ORBINP *)
```



```

FOR I:=1 TO 80 DO                                (* header *)
  IF NOT(EOLN(ORBINP)) THEN READ(ORBINP,HEADER[I]) ELSE HEADER[I]:=' ';
READLN(ORBINP);

FOR I := 1 TO 3 DO                                (* 3 observations *)
  BEGIN
    READ (ORBINP, YEAR, MONTH, DAY, UT);          (* date *)
    READ (ORBINP, D, M, S); DDD(D, M, S, RA[I]);  (* RA *)
    READLN(ORBINP, D, M, S); DDD(D, M, S, DEC[I]); (* Dec *)
    RA[I]:=RA[I]*15.0;
    JDO[I] := 2400000.5+MJD(DAY, MONTH, YEAR, UT);
    T[I] := (JDO[I]-2451545.0)/36525.0;
  END;
WRITELN;
READLN(ORBINP, EQX0); TEQX0:=(EQX0-2000.0)/100.0; (* equinox *)

(* desired equinox of the orbital elements *)

READ(ORBINP, EQX ); TEQX :=(EQX -2000.0)/100.0;

(* calculate initial data of the orbit determination *)

PMATECL(TEQX0, TEQX, A);
FOR I := 1 TO 3 DO
  BEGIN
    CART (1.0, DEC[I], RA[I], E[I, X], E[I, Y], E[I, Z]);
    EQUECL ( TEQX0 , E[I, X], E[I, Y], E[I, Z]);
    PRECART( A , E[I, X], E[I, Y], E[I, Z]);
    POLAR (E[I, X], E[I, Y], E[I, Z], DUMMY, BP[I], LP[I]);
    PMATECL( T[I], TEQX, AS);
    SUN200 (T[I], LS[I], BS[I], RS[I]);
    CART (RS[I], BS[I], LS[I], RSUN[I, X], RSUN[I, Y], RSUN[I, Z]);
    PRECART( AS , RSUN[I, X], RSUN[I, Y], RSUN[I, Z]);
  END;

WRITELN(' ORBDET: orbit determination from three observations ');
WRITELN(' version 93/07/01 ');
WRITELN(' (c) 1993 Thomas Pfleger, Oliver Montenbruck ');
WRITELN; WRITELN;
WRITELN(' Summary of orbit determination ');
WRITELN;
WRITE (' '); FOR I:=1 TO 78 DO WRITE(HEADER[I]); WRITELN;
WRITELN;
WRITELN(' Initial data (ecliptic geocentric coordinates (in deg))');
WRITELN;
WRITELN(' Julian Date ', JDO[1]:12:2, JDO[2]:12:2, JDO[3]:12:2);
WRITELN(' Solar longitude ', LS[1]:12:2, LS[2]:12:2, LS[3]:12:2);
WRITELN(' Planet/Comet Longitude', LP[1]:9:2, LP[2]:12:2, LP[3]:12:2);
WRITELN(' Planet/Comet Latitude ', BP[1]:9:2, BP[2]:12:2, BP[3]:12:2);
WRITELN; WRITELN;

```

END;

```
(*-----*)
(* DUMPELEM: output of orbital elements (screen) *)
(*-----*)
```

```
PROCEDURE DUMPELEM(TP,Q,ECC,INC,LAN,AOP,TEQX:REAL);
```

```
VAR DAY,MONTH,YEAR: INTEGER;
```

```
MODJD,UT : REAL;
```

```
BEGIN
```

```
MODJD := TP*36525.0 + 51544.5;
```

```
CALDAT( MODJD, DAY,MONTH,YEAR,UT);
```

```
WRITELN(' Orbital elements',
```

```
      ' (Equinox ', 'J', 100.0*TEQX+2000.0:8:2, '))');
```

```
WRITELN;
```

```
WRITELN(' Perihelion date      tp      ',
```

```
      YEAR:4, '/', MONTH:2, '/', DAY:2, UT:8:4, 'h',
```

```
      ' (JD', MODJD+2400000.5:11:2, '))');
```

```
WRITELN(' Perihelion distance q[AU] ', Q:12:6);
```

```
WRITELN(' Semi-major axis a[AU] ', Q/(1-ECC):12:6);
```

```
WRITELN(' Eccentricity e ', ECC:12:6);
```

```
WRITELN(' Inclination i ', INC:10:4, ' degrees');
```

```
WRITELN(' Ascending node Omega ', LAN:10:4, ' degrees');
```

```
WRITELN(' Long. of perihelion pi ', AOP+LAN:10:4, ' degrees');
```

```
WRITELN(' Arg. of perihelion omega ', AOP:10:4, ' degrees');
```

```
WRITELN;
```

```
END;
```

```
(*-----*)
(* SAVEELEM: output of orbital elements (file) *)
(*-----*)
```

```
PROCEDURE SAVEELEM(TP,Q,ECC,INC,LAN,AOP,TEQX:REAL;HEADER: CHAR80);
```

```
VAR I, DAY, MONTH, YEAR: INTEGER;
```

```
MODJD, UT : REAL;
```

```
BEGIN
```

```
(* open file for writing *)
```

```
(* REWRITE(ORBOUT); *)
```

```
(* Standard Pascal *)
```

```
ASSIGN(ORBOUT, 'ORBOUT.DAT'); REWRITE(ORBOUT);
```

```
(* Turbo Pascal *)
```

```
(* REWRITE(ORBOUT, 'ORBOUT.DAT'); *)
```

```
(* ST Pascal plus *)
```

```
MODJD := TP*36525.0 + 51544.5;
```

```
CALDAT( MODJD, DAY, MONTH, YEAR, UT);
```

```
WRITE (ORBOUT, YEAR:5, MONTH:3, (DAY+UT/24.0):7:3, '!' :6);
```

```
WRITELN(ORBOUT, ' perihelion time TO (y m d.d) = JD ',
```

```
      (MODJD+2400000.5):12:3);
```

```
WRITELN(ORBOUT, Q :12:6, '!' : 9, ' q ( a =', Q/(1-ECC):10:6, ' )');
```

```
WRITELN(ORBOUT, ECC:12:6, '!' : 9, ' e ');
```

```
WRITELN(ORBOUT, INC:10:4, '!' :11, ' i ');
```

```
WRITELN(ORBOUT, LAN:10:4, '!' :11, ' long.asc.node ');
```

```
WRITELN(ORBOUT, AOP:10:4, '!' :11,
```

```

      ' arg.perih. ( long.per. = ',AOP+LAN:9:4,' '));
WRITELN(ORBOUT,TEQI*100.0+2000.0:8:2,'!':13,' equinox (J)');
WRITE (ORBOUT,'! ');
FOR I:=1 TO 78 DO WRITE(ORBOUT,HEADER[I]);

RESET(ORBOUT); (* close file *)

END;

(*-----*)
(* RETARD: light-time correction *)
(* JDO: times of observation (t1',t2',t3') (Julian Date) *)
(* RHO: three geocentric distances (in AU) *)
(* JD: times of light emittance (t1,t2,t3) (Julian Date) *)
(* TAU: scaled time differences *)
(*-----*)
PROCEDURE RETARD ( JDO,RHO: REAL3; VAR JD,TAU: REAL3);
CONST KGAUSS = 0.01720209895; A = 0.00578;
VAR I: INTEGER;
BEGIN
  FOR I:=1 TO 3 DO JD[I]:=JDO[I]-A*RHO[I];
  TAU[1] := KGAUSS*(JD[3]-JD[2]); TAU[2] := KGAUSS*(JD[3]-JD[1]);
  TAU[3] := KGAUSS*(JD[2]-JD[1]);
END;

(*-----*)
(* GAUSS: iteration of the abbreviated Gauss method *)
(* *)
(* RSUN: three vectors of geocentric Sun positions *)
(* E : three unit vectors of geocentric observation directions *)
(* JDO : three observation times (Julian Date) *)
(* TP : time of perihelion passage (Julian centuries since J2000) *)
(* Q : perihelion distance *)
(* ECC : eccentricity *)
(* INC : inclination *)
(* LAN : longitude of the ascending node *)
(* AOP : argument of perihelion *)
(*-----*)
PROCEDURE GAUSS ( RSUN,E: MAT3X; JDO:REAL3;
VAR TP,Q,ECC,INC,LAN,AOP: REAL );

CONST EPS_RHO =1.0E-8;

VAR I,J : INTEGER;
S : INDEX;
RHOOLD,DET : REAL;
JD,RHO,N,TAU,ETA : REAL3;
DI : VECTOR;
RPL : MAT3X;
DD : REAL33;

```

```

BEGIN

```

```

(* calculate initial approximations of n1 and n3 *)
N[1] := (JD0[3]-JD0[2]) / (JD0[3]-JD0[1]);      N[2] := -1.0;
N[3] := (JD0[2]-JD0[1]) / (JD0[3]-JD0[1]);

(* calculate matrix D and its determinant (det(D) = e3.d3) *)
CROSS(E[2],E[3],DI); FOR J:=1 TO 3 DO DD[1,J]:=DOT(DI,RSUN[J]);
CROSS(E[3],E[1],DI); FOR J:=1 TO 3 DO DD[2,J]:=DOT(DI,RSUN[J]);
CROSS(E[1],E[2],DI); FOR J:=1 TO 3 DO DD[3,J]:=DOT(DI,RSUN[J]);
DET := DOT(E[3],DI);

WRITELN; WRITELN(' Iteration of the geocentric distances rho [AU] ');
WRITELN;

(* Iterate until distance rho[2] does not change any more *)

RHO[2] := 0;

REPEAT

  RHOOLD := RHO[2];

  (* geocentric distance rho from n1 and n3 *)
  FOR I := 1 TO 3 DO
    RHO[I]:=( N[1]*DD[I,1] - DD[I,2] + N[3]*DD[I,3] ) / (N[I]*DET);
  (* apply light-time correction and calculate time differences *)
  RETARD (JD0,RHO,JD,TAU);
  (* heliocentric coordinate vectors *)
  FOR I := 1 TO 3 DO
    FOR S := X TO Z DO
      RPL[I,S] := RHO[I]*E[I,S]-RSUN[I,S];

  (* sector/triangle ratios eta[i] *)
  ETA[1] := FIND_ETA( RPL[2], RPL[3], TAU[1] );
  ETA[2] := FIND_ETA( RPL[1], RPL[3], TAU[2] );
  ETA[3] := FIND_ETA( RPL[1], RPL[2], TAU[3] );

  (* improvement of the sector/triangle ratios *)
  N[1] := ( TAU[1]/ETA[1] ) / (TAU[2]/ETA[2]);
  N[3] := ( TAU[3]/ETA[3] ) / (TAU[2]/ETA[2]);
  WRITELN(' rho', ' ':16,RHO[1]:12:8,RHO[2]:12:8,RHO[3]:12:8);

UNTIL ( ABS(RHO[2]-RHOOLD) < EPS_RHO );

WRITELN; WRITELN(' Heliocentric distances [AU]:'); WRITELN;
WRITELN(' r ', ' ':16,
  NORM(RPL[1]):12:8,NORM(RPL[2]):12:8,NORM(RPL[3]):12:8);
WRITELN; WRITELN;

(* derive orbital elements from first and third observation *)
ELEMENT ( JD[1],JD[3],RPL[1],RPL[3], TP,Q,ECC,INC,LAN,AOP );

END;
```

(*-----*)

BEGIN

```
START(HEADER,RSUN,E,JDO,TEQX);
GAUSS(RSUN,E,JDO,TP,Q,ECC,INC,LAN,AOP);

DUMPELEM(TP,Q,ECC,INC,LAN,AOP,TEQX);
SAVEELEM(TP,Q,ECC,INC,LAN,AOP,TEQX,HEADER);
```

(* check solution *)

```
WRITELN;
IF (DOT(E[2],RSUN[2])>0) THEN
  WRITELN (' Warning: observation in hemisphere of conjunction;',
           ' possible second solution');
IF (ECC>1.1) THEN
  WRITELN (' Warning: probably not a realistic solution (e>1.1) ');
IF ( ( ABS(Q-0.985)<0.1) AND (ABS(ECC-0.015)<0.05) ) THEN
  WRITELN (' Warning: probably Earth's orbit solution');
```

END.

(*-----*)

Once ORBDET is called, no input is required from the user, because all the necessary information is held in the file ORBINP(.DAT) For the values given above we obtain the following report:

ORBDET: orbit determination from three observations
version 93/07/01
(c) 1993 Thomas Pflieger, Oliver Montenbruck

Summary of orbit determination

Ceres (Gauss's example)

Initial data (ecliptic geocentric coordinates (in deg))

Julian Date	2380570.51	2380704.42	2380830.35
Solar Longitude	162.91	297.21	61.95
Planet/Comet Longitude	95.54	99.82	118.10
Planet/Comet Latitude	-0.99	7.28	7.65

Iteration of the geocentric distances rho [AU]

rho	3.17388654	1.65960584	3.26891774
rho	2.97212839	1.81606981	3.04726038
rho	2.92283402	1.63070368	2.98636286
rho	2.90839087	1.63497331	2.96755085
rho	2.90390666	1.63628877	2.96157698
rho	2.90248678	1.63670369	2.95966629
rho	2.90203220	1.63683587	2.95905411

rho	2.90188709	1.63687808	2.95885777
rho	2.90184056	1.63689162	2.95879486
rho	2.90182566	1.63689595	2.95877468
rho	2.90182089	1.63689734	2.95876820
rho	2.90181936	1.63689778	2.95876612
rho	2.90181886	1.63689792	2.95876545
rho	2.90181871	1.63689797	2.95876524
rho	2.90181866	1.63689798	2.95876517
rho	2.90181864	1.63689799	2.95876515

Heliocentric distances [AU]:

r	2.68083158	2.58787040	2.54398482
---	------------	------------	------------

Orbital elements (Equinox J 1806.00)

Perihelion date	tp	1806/ 6/28 0.9118h	(JD 2380865.54)
Perihelion distance	q[AU]	2.541677	
Semi-major axis	a[AU]	2.767167	
Eccentricity	e	0.081488	
Inclination	i	10.6178 degrees	
Ascending node	Omega	80.9788 degrees	
Long. of perihelion	pi	147.0171 degrees	
Arg. of perihelion	omega	66.0383 degrees	

In addition, the orbital elements are output to the file ORBOUT.DAT:

```

1806 6 28.038      ! perihelion time T0 (y m d.d) = JD 2380865.538
      2.541677      ! q ( a = 2.767167 )
      0.081488      ! e
      10.6178       ! i
      80.9788       ! long.asc.node
      66.0383       ! arg.perih. ( long.per. = 147.0171 )
1806.00           ! equinox (J)
! Ceres (Gauss's example)

```

This may be used as input to the COMET program (see Chap. 4) in order to ensure that the observed positions are correctly reproduced:

Date	ET	Sun	l	b	r	RA	Dec	Distance
						h m s	o ' "	(AU)
1805/ 9/ 6	0.2	162.9	75.2	-1.1	2.681	6 23 57.5	22 21 27	2.901867
1806/ 1/17	22.1	297.2	106.4	4.6	2.588	6 45 14.7	30 21 24	1.636882
1806/ 5/23	20.4	61.9	137.7	8.9	2.544	8 7 44.6	28 2 47	2.958699

As can be seen from the ecliptic coordinates of the Sun and Ceres, the minor planet was close to opposition at the time of observation. In the example chosen, there is no danger of a double solution to the orbit determination.

12. Astrometry

Photography provides a relatively simple method of determining the coordinates of a comet or planet with respect to the known positions of stars. All that is required apart from the plate itself is a stellar catalogue and possibly an atlas to make it easier to identify the neighbouring stars.

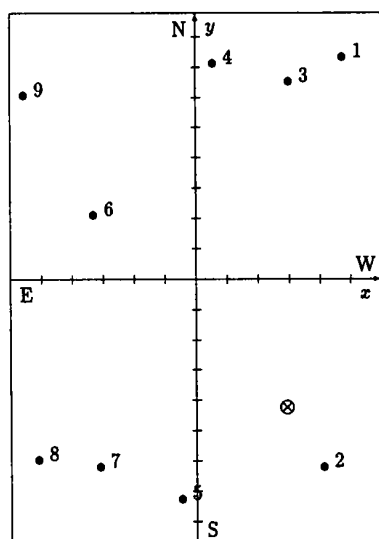
An actual photograph of Comet Bradfield in 1982 is shown diagrammatically in Fig. 12.1 to help in the following discussion. Comparison with a star atlas shows that the picture covers an area $\pm 4^m$ in right ascension and $\pm 1^\circ$ in declination, and that it is centred on the point ($\alpha_{1950} = 12^h 26^m 15^s$, $\delta_{1950} = 44^\circ 25'$). Apart from the comet, a number of stars can be identified, whose coordinates can be obtained from the Smithsonian Astrophysical Observatory's *SAO Catalog*. This covers the whole of the northern and southern skies and contains about 250 000 stars. The average distance between two neighbouring stars is about half a degree. At least three reference stars with known coordinates are required to determine the position of a celestial object from a photograph. With a view to obtaining the most accurate results, we nevertheless usually try to determine the required coordinates using the largest possible number of neighbouring stars. First, we will discuss how a section of the sky containing stars of different right ascensions and declinations will be reproduced on the plate.

12.1 Photographic Imaging

An ideal camera or telescope objective brings the rays of light from a star to a point on the film-plane. This lies at a distance F , known as the focal length, behind the objective. The point P , at which the star is imaged, can be found by the projection of the ray of light through the centre of the objective O (Fig. 12.2).

In a coordinate system defined by u , v , and w ,

$$\mathbf{e} = \begin{pmatrix} \cos(\delta) \cos(\alpha - \alpha_0) \\ \cos(\delta) \sin(\alpha - \alpha_0) \\ \sin(\delta) \end{pmatrix}$$



Comet Bradfield on 1982 Sept. 4

Reference stars:

- 1 SAO 044166
- 2 SAO 044175
- 3 SAO 044177
- 4 SAO 044187
- 5 SAO 044199
- 6 SAO 044207
- 7 SAO 044208
- 8 SAO 044220
- 9 SAO 044221

⊗ Comet

Plate centre:

$$\alpha \approx 12^{\text{h}}26^{\text{m}}15^{\text{s}}$$

$$\delta \approx 44^{\circ}15'$$

Fig. 12.1. An example of a star field

is the direction of a star at right ascension α and declination δ . Correspondingly,

$$\mathbf{e}_0 = \begin{pmatrix} \cos(\delta_0) \\ 0 \\ \sin(\delta_0) \end{pmatrix}$$

defines the point (α_0, δ_0) on the sky at which the axis of the camera is pointing. The vectors $\mathbf{F} = -\mathbf{F} \cdot \mathbf{e}_0$ and $\mathbf{p} = -\mathbf{p} \cdot \mathbf{e}$ describe the paths followed by a ray of light from the objective to the centre of the plate, and to the image P of the star. They include the angle φ , where

$$\begin{aligned} \cos(\varphi) &= \mathbf{e}_0 \cdot \mathbf{e} \\ &= \cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta) \end{aligned}$$

Within the film-plane, the vectors

$$\mathbf{e}_X = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{e}_Y = \begin{pmatrix} +\sin(\delta_0) \\ 0 \\ -\cos(\delta_0) \end{pmatrix}$$

define a system of coordinates that is used to measure the plate, and which is oriented North-South and East-West. If the coordinates X and Y of point P are measured in units of the focal length F , $\mathbf{p}$ may be expressed as:

$$\mathbf{p} = \mathbf{F} + (F \cdot X) \cdot \mathbf{e}_X + (F \cdot Y) \cdot \mathbf{e}_Y$$

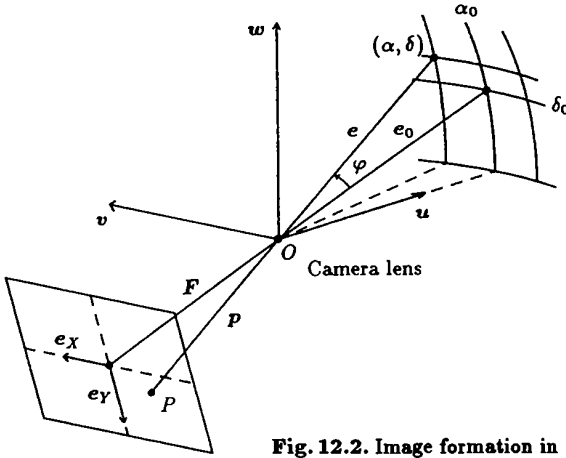


Fig. 12.2. Image formation in photographing a star field

If we write this equation in component form, the relationship between (α, δ) on one side and (X, Y) on the other can be expressed by the three equations

$$\begin{aligned} p \cos(\delta) \cos(\alpha - \alpha_0) &= F \cos(\delta_0) - FY \sin(\delta_0) \\ p \cos(\delta) \sin(\alpha - \alpha_0) &= -FX \\ p \sin(\delta) &= F \sin(\delta_0) + FY \cos(\delta_0) \end{aligned} \quad (12.1)$$

where

$$p = |p| = F\sqrt{1 + X^2 + Y^2}$$

or

$$\begin{aligned} p &= F / \cos(\varphi) \\ &= F / (\cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta)) \end{aligned}$$

Solving for the spherical coordinates gives us the equations

$$\begin{aligned} \alpha &= \alpha_0 + \arctan \left\{ \frac{-X}{\cos(\delta_0) - Y \sin(\delta_0)} \right\} \\ \delta &= \arcsin \left\{ \frac{\sin(\delta_0) + Y \cos(\delta_0)}{\sqrt{1 + X^2 + Y^2}} \right\} \end{aligned} \quad (12.2)$$

Rearranging (12.2) we can similarly obtain

$$\begin{aligned} X &= -\frac{\cos(\delta) \sin(\alpha - \alpha_0)}{\cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta)} \\ Y &= -\frac{\sin(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) - \cos(\delta_0) \sin(\delta)}{\cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta)} \end{aligned} \quad (12.3)$$

from (12.1).

Both of these transforms can be simply expressed in sub-routines:

```

(*-----*)
(* STDEQU: transformation from standard coordinates into          *)
(*      equatorial coordinates                                    *)
(*  RAO,DECO: right ascension and declination of the optical axis (deg) *)
(*  XX,YY:      standard coordinates                              *)
(*  RA,DEC:      right ascension and declination (deg)           *)
(*-----*)
PROCEDURE STDEQU ( RAO,DECO,XX,YY: REAL; VAR RA,DEC: REAL);
  BEGIN
    RA := RAO + ATN ( -XX / (CS(DECO)-YY*SN(DECO)) );
    DEC := ASN ( (SN(DECO)+YY*CS(DECO))/SQRT(1.0+XX*XX+YY*YY) );
  END;
(*-----*)
(* EQUSTD: transformation of equatorial coordinates into          *)
(*      standard coordinates                                       *)
(*  RAO,DECO: right ascension and declination of the optical axis (deg) *)
(*  RA,DEC:      right ascension and declination (deg)           *)
(*  XX,YY:      standard coordinates                              *)
(*-----*)
PROCEDURE EQUSTD ( RAO,DECO,RA,DEC: REAL; VAR XX,YY: REAL);
  VAR C: REAL;
  BEGIN
    C := CS(DECO)*CS(DEC)*CS(RA-RAO)+SN(DECO)*SN(DEC);
    XX := - ( CS(DEC)*SN(RA-RAO) ) / C;
    YY := - ( SN(DECO)*CS(DEC)*CS(RA-RAO)-CS(DECO)*SN(DEC) ) / C;
  END;
(*-----*)

```

12.2 Plate Constants

The dimensionless coordinates X and Y are described as standard coordinates, because, by definition, they are not dependent on the focal length of the optics employed, and refer to a system of coordinates that is oriented exactly parallel to the meridian that passes through the centre $((\alpha_0, \delta_0)$ of the plate. If this coordinate system is also to be used as the basis for the plate reduction, then the measured coordinates x and y only have to be divided by the focal length to obtain the standard coordinates:

$$X = x/F \quad Y = y/F \quad .$$

In general, the origin of the coordinate system used will, nevertheless, not precisely coincide with the intersection of the optical axis with the film-plane. An offset of this nature, amounting to $(\Delta x, \Delta y)$ can, however, be accommodated by a small alteration to the equation just given:

$$X = x/F - (\Delta x)/F \quad Y = y/F - (\Delta y)/F \quad .$$

If, in addition, the coordinate axes are rotated through an angle γ with respect to the North-South line, then the transformation equations become:

$$X = (x \cdot \cos(\gamma) - y \cdot \sin(\gamma))/F - (\Delta x)/F$$

$$Y = (x \cdot \sin(\gamma) + y \cdot \cos(\gamma))/F - (\Delta y)/F .$$

However, offset and rotation of the coordinate system are not the only factors that affect the relationship between the measured coordinates and the standard coordinates. Optical errors, or possible tilt or distortion of the film generally mean that further corrections have to be made. In general, therefore, the equations are expressed in the form:

$$X = a \cdot x + b \cdot y + c \quad (12.4)$$

$$Y = d \cdot x + e \cdot y + f$$

with six *plate constants* a, b, c, d, e , and f , allowing conversion between the (x, y) and (X, Y) coordinate pairs. The plate constants cannot be known in advance, but must be determined by the use of reference stars. If we know the equatorial coordinates $(\alpha_i, \delta_i)_{i=1,2,3}$ of three reference stars, then from (12.3), we obtain the three equations

$$\begin{aligned} X_1 &= x_1 \cdot a + y_1 \cdot b + c \\ X_2 &= x_2 \cdot a + y_2 \cdot b + c \end{aligned} \quad (12.5)$$

$$X_3 = x_3 \cdot a + y_3 \cdot b + c ,$$

which may be solved for a, b , and c by appropriate manipulation. The remaining plate constants d, e , and f are correspondingly obtained from the equations for Y_i . The focal length of the camera optics is not required. This is of particular advantage when one is working with an enlargement whose scale is unknown, rather than with an original plate.

12.3 Reduction

As we have just seen, the plate constants may be determined by solving two equations, each with three unknowns, if at least three reference stars with known right ascension and declination are available. As the measurement of stellar coordinates can never be completely free from errors, it is desirable to use as many reference stars as possible in determining the plate constants. Normally, however, a set of equations of the form

$$\begin{aligned} X_1 &= x_1 \cdot a + y_1 \cdot b + c \\ &\dots \end{aligned} \quad (12.6)$$

$$X_n = x_n \cdot a + y_n \cdot b + c ,$$

with $n > 3$ equations for the three desired plate constants is overdetermined and can therefore no longer be solved. To avoid this problem, the reduction procedure about to be described uses the *method of least squares*.

In general, if we want to determine m unknowns $s_1, \dots, s_m$ from n equations

$$t_i = A_{i1} \cdot s_1 + \dots + A_{im} \cdot s_m \quad (i = 1, \dots, n) \quad (12.7)$$

Using similar rearrangements, which are also known as Givens rotations, one after the other $A_{31} \dots A_{n1}$, $A_{32} \dots A_{n2}$, ..., and $A_{m+1,m} \dots A_{nm}$ are also eliminated, until finally we obtain a system of equations in the form of (12.9). The sum of the residuals squared can now be written as two terms

$$S = \sum_{i=1}^m r_i'^2 + \sum_{i=m+1}^n t_i'^2 ,$$

of which only the first is still dependent on the unknown s_k . The least sum will obviously be obtained when all values of r_i disappear for $i = 1 \dots m$. The corresponding values of s_k are obtained as follows:

$$s_m = t'_m / A'_{mm} \quad (12.10)$$

$$s_k = (t'_k - \sum_{l=k+1}^m A'_{kl} s_l) / A'_{kk} \quad (k = m-1, \dots, 1) .$$

The rearrangement of the set of equations and the subsequent solution are combined in the LSQFIT procedure, which provides an easy method of solving any linear problem of this sort.

```
(*-----*)
(* LSQFIT: *)
(* solution of an overdetermined system of linear equations *)
(* A[i,1]*s[1]+...A[i,m]*s[m] - A[i,m+1] = 0 (i=1,...,n) *)
(* according to the method of least squares using Givens rotations *)
(* A: matrix of coefficients *)
(* N: number of equations (rows of A) *)
(* M: number of unknowns (M+1=columns of A, M=elements of S) *)
(* S: solution vector *)
(*-----*)
```

```
PROCEDURE LSQFIT ( A: LSQMAT; N, M: INTEGER; VAR S: LSQVEC );
```

```
CONST EPS = 1.0E-10; (* machine accuracy *)
```

```
VAR I,J,K: INTEGER;
```

```
P,Q,H: REAL;
```

```
BEGIN
```

```
FOR J:=1 TO M DO (* loop over columns 1...M of A *)
```

```
(* eliminate matrix elements A[i,j] with i>j from column j *)
```

```
FOR I:=J+1 TO N DO
```

```
IF A[I,J]<>0.0 THEN BEGIN
```

```
(* calculate p, q and new A[j,j]; set A[i,j]=0 *)
```

```
IF ( ABS(A[J,J])<EPS*ABS(A[I,J]) )
```

```
THEN
```

```
BEGIN
```

```
P:=0.0; Q:=1.0; A[J,J]:=-A[I,J]; A[I,J]:=0.0;
```

```
END
```

```
ELSE
```

```
BEGIN
```

```

      H:=SQRT(A[J,J]*A[J,J]+A[I,J]*A[I,J]);
      IF A[J,J]<0.0 THEN H:=-H;
      P:=A[J,J]/H; Q:=-A[I,J]/H; A[J,J]:=H; A[I,J]:=0.0;
    END;
    (* calculate rest of the line *)
    FOR K:=J+1 TO M+1 DO
      BEGIN
        H      := P*A[J,K] - Q*A[I,K];
        A[I,K] := Q*A[J,K] + P*A[I,K];
        A[J,K] := H;
      END;
    END;
    (* backsubstitution *)
    FOR I:=M DOWNT0 1 DO
      BEGIN
        H:=A[I,M+1];
        FOR K:=I+1 TO M DO H:=H-A[I,K]*S[K];
        S[I] := H/A[I,I];
      END;
    END;
  END; (* LSQFIT *)

(*-----*)

```

The array **A** stores the coefficients A_{ij} ($i = 1 \dots n, j = 1 \dots m$) in the first m columns, and the values of t_i ($i = 1 \dots n$) in another column. After the procedure is called, **S** contains the desired values for the unknowns $s_1 \dots s_m$. In working with the LSQFIT procedure the two types

```

  TYPE  LSQVEC = ARRAY[1..MDIM] OF REAL;
        LSQMAT = ARRAY[1..NDIM,1..M1DIM] OF REAL;

```

must be declared. The dimensions **MDIM** and **NDIM** should be chosen to be at least as large as the number of unknowns (m) and equations (n) respectively. **M1DIM** gives the number of columns of **A** and should not, therefore be less than $m + 1$.

It should also be noted that storage of the matrix **A** is not absolutely necessary in the procedure described for solving for the best fit. The individual coefficients A_{ij} may be eliminated by rows, rather than by columns. By appropriate programming, large problems with many equations can thus be handled efficiently. Nevertheless, the version of LSQFIT given here is somewhat easier to use, and is quite adequate for the task in mind.

12.4 The FOTO Program

The FOTO program enables accurate positions of stars, comet, or minor planets to be obtained from photographic plates of the sky. It will relieve the user of the considerable amount of computational work that is required in determining standard coordinates, and solving for the residuals.

Before calling FOTO, however, a certain amount of preparation is required to obtain the necessary input data. First, using a star chart, a few of the brighter stars must be identified, thus establishing the approximate coordinates of the field covered by the plate. Then suitable reference stars are chosen, whose equatorial coordinates can be found in a star catalogue, such as the *SAO Catalog*. If necessary, the proper motions of the stars between the epoch of the catalogue and the date of the exposure must be taken into account. The reference stars should be evenly distributed across the plate, and should be as point-like as possible, to make it easier to obtain accurate measurements. The positions of the reference stars and the object being examined can be determined without difficulty, by laying a transparent millimetre grid, oriented approximately North-South, over an enlargement. Finally, we determine the right ascension and declination of the plate centre, which are required in calculating the standard coordinates. As errors in the coordinates of the plate centre have little effect on position determination, it suffices to take these values from an atlas.

The various pieces of data are all entered in a data file FOTINP(.DAT). Its first line contains the right ascension (in $^h, ^m, ^s$) and declination (in $^\circ, ', ''$) of the plate centre. This is followed by details of the individual objects on the plate. Reference stars that are to be used to determine the plate constants are indicated by an asterisk (*) in the first column. After the name, which may contain up to 11 letters, the measured coordinates (x, y in mm) are given, and additionally for reference stars, their equatorial coordinates (α in $^h, ^m, ^s$, δ in $^\circ, ', ''$). The data given in the following example refer to the plate shown diagrammatically in Fig. 12.1.

CENTRE			12 26 15.0	44 15 00.0
*SAO 044166	+47.5	+73.3	12 22 40.293	45 04 33.40
*SAO 044175	+41.2	-62.1	12 23 32.960	43 23 42.59
*SAO 044177	+29.9	+65.2	12 23 56.037	44 59 16.14
*SAO 044187	+5.5	+71.1	12 25 39.308	45 04 14.33
*SAO 044199	-4.8	-72.7	12 26 43.594	43 17 28.86
*SAO 044207	-33.0	+21.0	12 28 27.655	44 27 53.23
*SAO 044208	-31.2	-62.0	12 28 27.755	43 26 04.19
*SAO 044220	-51.2	-59.7	12 29 52.083	43 28 33.76
*SAO 044221	-55.7	+60.6	12 29 56.524	44 57 27.65
BRADFIELD	+29.2	-42.1		

The procedure GETINP reads and stores these values. Then the standard coordinates of the reference stars, which are required to evaluate the plate constants, are determined. From the plate constants, the procedure can be reversed: the standard coordinates and equatorial coordinates of all the objects on the plate being determined from the measured coordinates. With the reference stars, this enables one to obtain a good estimate of the error that one can expect in reducing the plate. In addition, any gross errors can be recognized, such as individual reference stars having been incorrectly identified or measured.

```

(*-----*)
(*                                FOTO                                *)
(*      astrometric analysis of photographic plates                  *)
(*                                version 93/07/01                    *)
(*-----*)
PROGRAM FOTO (INPUT,OUTPUT,FOTINP);

  CONST MAXDIM      = 30;      (* maximum number of objects on the foto *)
  NAME_LENGTH = 12;
  ARC              = 206264.8; (* arcseconds per radian *)

  TYPE NAME_TYPE = ARRAY[1..NAME_LENGTH] OF CHAR;
  LSQVEC = ARRAY[1..3] OF REAL;
  LSQMAT = ARRAY[1..MAXDIM,1..5] OF REAL;
  REAL_ARRAY = ARRAY[1..MAXDIM] OF REAL;
  NAME_ARRAY = ARRAY[1..MAXDIM] OF NAME_TYPE;

  VAR I,J,K, NREF,NOBJ, DEG,MIN : INTEGER;
  RAO,DECO, A,B,C,D,E,F, SEC : REAL;
  RA_OBS,DEC_OBS, D_RA,D_OEC : REAL;
  DET, FOC_LEN, SCALE : REAL;
  RA,DEC, X,Y, XX,YY, DELTA : REAL_ARRAY;
  S : LSQVEC;
  AA : LSQMAT;
  NAME : NAME_ARRAY;
  FOTINP : TEXT;

(*-----*)
(* The following procedures are to be entered here in the order given *)
(* SN, CS, ATN, ASN, DDD, DMS, LSQFIT *)
(* EQUSTD, STDEQU *)
(*-----*)

(*-----*)
(* GETINP: read input data from file FOTINP *)
(*-----*)
PROCEDURE GETINP ( VAR RAO,DECO: REAL; VAR NOBJ: INTEGER;
  VAR NAME: NAME_ARRAY; VAR RA,DEC,X,Y: REAL_ARRAY );
  VAR I,K,H,M: INTEGER;
  S : REAL;
  C : CHAR;
BEGIN
  WRITELN;
  WRITELN ( '      FOTO: astrometric analysis of photographic plates ');
  WRITELN ( '                                version 93/07/01                ');
  WRITELN ( '      (c) 1993 Thomas Pflieger, Oliver Montenbruck ');
  WRITELN;

  (* open file for reading *
  (* RESET(FOTINP); *) (* Standard Pascal *)
  ASSIGN(FOTINP,'FOTINP.DAT'); RESET(FOTINP); (* TURBO Pascal *)
  (* RESET(FOTINP,'FOTINP.DAT'); *) (* ST Pascal plus *)

```



```

WRITELN (' Input data file: FOTINP'); WRITELN;

(*read coordinates of the plate center *)
FOR K:=1 TO NAME_LENGTH DO READ(FOTINP,C);
READ (FOTINP,H,M,S); DDD(H,M,S,RA0); RA0:=15.0*RA0;
READLN(FOTINP,H,M,S); DDD(H,M,S,DECO);

(* read name, plate coordinates (and equatorial coordinates) *)
I := 0;
REPEAT
  I := I+1;
  FOR K:=1 TO NAME_LENGTH DO READ(FOTINP,NAME[I][K]);  (* name *)
  IF NAME[I][1]='*'
    THEN (* reference star *)
      BEGIN
        READ (FOTINP,X[I],Y[I]);
        READ (FOTINP,H,M,S); DDD(H,M,S,RA[I]); RA[I]:=15.0*RA[I];
        READLN(FOTINP,H,M,S); DDD(H,M,S,DEC[I]);
      END
    ELSE (* unknown object *)
      BEGIN
        READLN (FOTINP,X[I],Y[I]); RA[I]:=0.0; DEC[I]:=0.0;
      END;
  UNTIL EOF(FOTINP);

  NOBJ := I;

END;

(*-----*)

BEGIN (* FOTO *)

  (* read input from file *)

  GETINP ( RAO,DECO, NOBJ, NAME, RA,DEC,X,Y );

  (* calculate standard coordinates of reference stars; *)
  (* fill elements of matrix AA (column AA[*],5 serves *)
  (* as intermediate storage) *)

  J:=0;
  FOR I:=1 TO NOBJ DO
    IF NAME[I][1]='*' THEN
      BEGIN
        J := J+1;
        EQUSTD ( RAO,DECO, RA[I],DEC[I], XX[I],YY[I] );
        AA[J,1]:= X[I]; AA[J,2]:= Y[I]; AA[J,3]:=1.0;
        AA[J,4]:=+XX[I]; AA[J,5]:=+YY[I];
      END;
  WREF := J;  (* number of reference stars *)

```

```

(* calculate plate constants a,b,c *)

LSQFIT ( AA, NREF, 3, S );  A:=S[1]; B:=S[2]; C:=S[3];

(* calculate plate constants d,e,f *)

FOR I:=1 TO NREF DO  AA[I,4]:=AA[I,5]; (* copy column A[* ,5]->A[* ,4] *)
LSQFIT ( AA, NREF, 3, S );  D:=S[1]; E:=S[2]; F:=S[3];

(* calculate equatorial coordinates (and errors for reference stars) *)

FOR I:=1 TO NOBJ DO
  BEGIN
    XX[I] := A*X[I]+H*Y[I]+C;
    YY[I] := D*X[I]+E*Y[I]+F;
    STDEQU ( RAO,DECO, XX[I],YY[I], RA_OBS,DEC_OBS );
    IF NAME[I][1]='*' THEN (* find error in arcseconds *)
      BEGIN
        D_RA := (RA_OBS-RA[I])*CS(DEC[I]);
        D_DEC := (DEC_OBS-DEC[I]);
        DELTA[I] := 3600.0 * SQRT ( D_RA*D_RA + D_DEC*D_DEC );
      END;
    RA[I] := RA_OBS;  DEC[I] := DEC_OBS;
  END;

(* focal length *)

DET := A*E-D*B;
FOC_LEN := 1.0/SQRT(ABS(DET));
SCALE := ARC / FOC_LEN;

(* output *)

WRITELN ( ' Plate constants:' );
WRITELN;
WRITELN ( '  a =',a:12:8, '  b =',b:12:8, '  c =',c:12:8);
WRITELN ( '  d =',d:12:8, '  e =',e:12:8, '  f =',f:12:8);
WRITELN;
WRITELN ( ' Effective focal length and image scale:' );
WRITELN;
WRITELN ( '  F =',FOC_LEN:9:2, ' mm' );
WRITELN ( '  m =', SCALE:7:2, ' "/mm' );
WRITELN;
WRITELN ( ' Coordinates:' );
WRITELN;
WRITELN ( ' Name':11, ' x':9, ' y':7, ' X':8, ' Y':8,
           ' RA':12, ' Dec':13, ' Error':9 );
WRITELN ( ' mm':20, ' mm':7, ' '':23,
           ' h m s ', ' o ' ' " '':12, ' " '':6 );

FOR I:=1 TO NOBJ DO
  BEGIN
    WRITE( ' '); FOR K:=1 TO NAME_LENGTH DO WRITE(NAME[I][K]);

```

```

WRITE(X[I]:7:1,Y[I]:7:1,XX[I]:9:4,YY[I]:8:4);
DMS(RA[I]/15.0,DEG,MIN,SEC); WRITE(DEG:5,MIN:3,SEC:6:2);
DMS(DEC[I],DEG,MIN,SEC); WRITE(DEG:4,MIN:3,SEC:5:1);
IF NAME[I][1]='*' THEN WRITE(DELTA[I]:6:1);
WRITELN;
END;
WRITELN;

END. (* FOTO *)

(*-----*)

```

Using the data given earlier, FOTO produces the following output:

FOTO: astrometric analysis of photographic plates
version 93/07/01
(c) 1993 Thomas Pflieger, Oliver Montenbruck

Input data file: FOTINP

Plate constants:

a = 0.00021648 b = 0.00000791 c = 0.00013818
d = -0.00000789 e = 0.00021643 f = -0.00103320

Effective focal length and image scale:

F = 4616.85 mm
m = 44.68 "/mm

Coordinates:

Name	x mm	y mm	X	Y	RA h m s	Dec ° ' "	Error "
*SAO 044166	47.5	73.3	0.0110	0.0145	12 22 40.82	45 4 29.1	7.0
*SAO 044175	41.2	-62.1	0.0086	-0.0148	12 23 32.90	43 24 0.7	18.1
*SAO 044177	29.9	65.2	0.0071	0.0128	12 23 56.46	44 59 3.5	13.4
*SAO 044187	5.5	71.1	0.0019	0.0143	12 25 38.18	45 4 11.4	12.3
*SAO 044199	-4.8	-72.7	-0.0015	-0.0167	12 26 42.88	43 17 29.4	7.8
*SAO 044207	-33.0	21.0	-0.0068	0.0038	12 28 26.78	44 27 53.3	9.4
*SAO 044208	-31.2	-62.0	-0.0071	-0.0142	12 28 29.56	43 26 5.1	19.7
*SAO 044220	-51.2	-59.7	-0.0114	-0.0136	12 29 51.32	43 28 12.5	22.8
*SAO 044221	-55.7	60.6	-0.0114	0.0125	12 29 57.32	44 57 49.1	23.1
BRADFIELD	29.2	-42.1	0.0061	-0.0104	12 24 18.57	43 39 16.4	

After the plate constants, the effective focal length (= Camera focal length × Enlargement) and the image scale are displayed. These are followed by the measured rectangular coordinates (x, y) for each object, together with the standard coordinates (X, Y) calculated from them, and the equatorial coordinates right ascension and declination. The reference stars show typical departures from the catalogued places of between 10" to 20". This corresponds

to measurement errors of about $1/4$ – $1/2$ mm, which are to be expected in determining errors with millimetre grid overlays. More accurate results can generally be obtained by using a special measuring machine.

The proper motion during the interval between the epoch of the *SAO Catalog* and the time of the exposure (1982.8) was not taken into account. It amounts to $7''$ at most for the chosen reference stars, and is therefore smaller than the scatter found in determining the positions.

Appendix

A.1 Notes on Alterations to Suit Individual Computers

Although programs that are written in Pascal can generally be implemented easily on different computers, there are, however, some minor differences between different implementations of the language. One reason for this is the need to adapt the compiler to the machine and operating system architecture, in order to make optimum use of the hardware. The following section considers some of the most important problems that can arise in adapting the programs. More information can usually be found in the manuals that accompany the computer.

Integer Arithmetic

Particularly obvious differences occur from computer to computer with regard to the range of values that can be handled for **REAL** and **INTEGER** numbers, and the relative accuracy of **REAL** arithmetic. Both largely depend on the number of bytes¹ that are used to store a number. Most commonly encountered are 2- and 4-byte **INTEGER** numbers, which cover the range $-32767 \dots +32767$ and $-2147483647 \dots +2147483647$, respectively. Various compilers offer the use of both forms, because computations using 2-byte numbers are generally significantly faster. They are differentiated by the use of the type **LONG_INTEGER**, for example, in addition to **INTEGER**.

In the programs given here, it is generally unimportant which form of the type **INTEGER** is used to represent integers. The only exception is in obtaining the integer part of a decimal value, for which the standard function

```
FUNCTION TRUNC ( X: REAL): INTEGER;
```

is generally available. If the argument exceeds the range of the **INTEGER** type, this leads to a run-time error when the program is implemented. For this reason in the two procedures **CALDAT** and **LMST** it is absolutely essential to employ the function

```
FUNCTION LONG_TRUNC ( X: REAL): LONG_INTEGER;
```

¹1 byte = 8 bits. One bit is the smallest storage unit in a computer. It stores a single binary digit with a value of 0 or 1.

instead of **TRUNC**, if the compiler being used supports 2-byte **INTEGER** numbers by default. Not all versions of Turbo Pascal, which is a very widespread implementation of the language, provide 4-byte arithmetic, but all versions supply a function **FUNCTION INT (X: REAL): REAL**; which may be used in the programs mentioned instead of **TRUNC**.

The range of **INTEGER** numbers may, in principle, also be exceeded when using the **TRUNC** function in the **SUN200**, **MINI_MOON**, **NUTEQU** and **MOON** procedures, as well as within **IMPROVE** in the program **MOON**, although only in calculations for dates prior to the year -400. If problems occur, **TRUNC** may be replaced by **LONG_TRUNC** or **INT** here as well.

Floating-Point Arithmetic

For most of the programs discussed in this book, the accuracy of the calculations carried out with floating-point arithmetic is generally of secondary importance. This needs to be briefly explained. In calculating the coordinates of the Moon or of the planets, the basic model and the numerical constants permit us to obtain an accuracy of about 1", for example. This corresponds to a number being correct to 5-6 decimal places, to which another 2-3 places need to be added to allow for rounding errors that can occur in evaluating various equations. If we are working with 11-12 decimal-place arithmetic, which is available with all modern compilers running on personal computers, workstations, or main-frames, we have no need to worry about encountering any noticeable computational errors. This is also true for the numerical integration of minor planet orbits in the **NUMINT** program, unless one is interested in long-term integrations over many revolutions. In this case a 15-16 digit accuracy is recommended. Aside from this it is advisable not to work with less than 8-place accuracy, however, because this can cause problems in the routines used for calendar calculations.

In some procedures constants are employed that require a machine accuracy of $u \approx 10^{-11} \dots 10^{-12}$. Here u is the smallest positive number that yields a result different from 1.0 when added to 1.0. If necessary, it can be determined by the function **EPSMACH** for arbitrary compilers.

```
(*-----*)
(* EPSMACH computes the machine accuracy u (1.0+u>1.0,1.0+u/2=1.0) *)
(*-----*)
FUNCTION EPSMACH: REAL;
  VAR ONE,TWO,U: REAL;
  BEGIN
    ONE:=1.0; TWO:=2.0; U:=1.0;
    REPEAT U:=U/TWO; UNTIL ( (ONE+U)=ONE );
    EPSMACH := TWO*U;
  END;
(*-----*)
```

The dependent constants are collated in the following table.

Routine	Constant	Value
ASN	EPS =1.0E-7	$0.1\sqrt{u}$
ACS	EPS =1.0E-7	$0.1\sqrt{u}$
ECCANOM	EPS =1.0E-11	u
HYPANOM	EPS =1.0E-10	$10u$
STUMPPF	EPS =1.0E-12	$0.1u$
PARAB	EPS =1.0E-9	10^2u
DE	FOURU =8.0E-12	$4u$
STEP	TWOU =4.0E-12	$2u$
STEP	FOURU =8.0E-12	$4u$
FIND_ETA	DELTA =1.0E-9	10^2u
GAUSS	EPS_RHO=1.0E-8	10^3u
LSQFIT	EPS =1.0E-10	$10u$

These values may be adjusted to suit the available accuracy if required. A modification is advisable for DE and STEP to make full use of the available accuracy in the numerical integration. Aside from this, a change is only really necessary if the routines ECCANOM, HYPANOM, PARAB, FIND_ETA or GAUSS do not converge while using low computational accuracy.

With compilers that support single- or double-precision arithmetic, generally single-precision accuracy (ca. 7 places) is not sufficient. To employ the higher precision, however, other type declarations (for example DOUBLE or LONG_FLOAT instead of REAL) are required to carry out floating-point arithmetic. In most cases one can avoid having to make alterations throughout by including the global definition TYPE REAL=DOUBLE; at the beginning of the program. In addition, it should be noted that a series of literal constants must be specifically declared as double-precision. This generally means that they need to be written in exponential form, with E replaced by D, such as $PI = 3.141592654D0$ instead of $PI = 3.141592654$.

Data Files

In nearly all implementations of Pascal the handling of external data files has led to extensions to the basic standard, which is poorly suited to the data-handling used by most modern operating systems. The most important consequence is the relationship between the names of the file variables within a program and the external names that the operating system uses to refer to the data files. A program to read two numbers from a data file and then output them again would be written as follows in Standard-Pascal:

```
PROGRAM READIN ( INPUT, OUTPUT, F );
  VAR X,Y: REAL; F: TEXT;
  BEGIN
    RESET(F); READ(F,X,Y); WRITE(X,Y);
  END;
```

The file variable **F** is declared in the first line, because the corresponding data file is permanently available, and does not just exist while the program executes. In order to read the required data from a file with the name **DATA.DAT**, there are various variants of the **RESET** command, the most common of which are given in the following table.

Compiler	Commands to open the data file before a read
Turbo Pascal	ASSIGN(F, 'DATA.DAT'); RESET(F);
ST Pascal plus	RESET(F, 'DATA.DAT');
DEC Vax	OPEN(F, 'DATA.DAT'); RESET(F);

Similarly, data files may be opened before a write instruction by modifying the **REWRITE** command.

Procedures as Parameters

Pascal allows procedures and functions to be passed as parameters to sub-routines. Use of this is made in the routine **T_FIT_LBR**, which expands coordinates by Chebyshev polynomials. As some compilers (e.g. early versions of Turbo Pascal) do not allow this, the routines **T_FIT_MOON** and **T_FIT_SUN** must, if necessary, be rewritten. However, it is possible to do this without too much difficulty. It is only necessary to replace the instruction lines in the two procedures mentioned by the declaration and instruction lines of **T_FIT_LBR** and then to change the variable names **L_POLY** and **B_POLY** into **RA_POLY** and **DE_POLY**, as well as **L** and **B** into **RA** and **DE**. Correspondingly modified versions of **T_FIT_MOON** and **T_FIT_SUN** are included on the diskette of the programs in this book.

Variable Names

In the programs given in this book, the names of many variables and procedures contain an underscore (**_**). This is *not* in accordance with the ANSI or ISO standards. However, because of the greater clarity and legibility that can be obtained by using **_**, we decided not to forego its use.

As nowadays nearly all the compilers recognize and allow this extension to the permissible set of letters, this should cause few problems. If it does, the underscore can be replaced by some other letter (such as **X**).

A.2 List of Procedures

The following list provides a summary of the functions and procedures that are not linked to a specific main program and which are therefore particularly suitable for use in any new programs. Apart from a short description, the list gives the section in which the procedure's listing is given, and a note of the procedures that it calls.

ABERRAT	10.1	stellar aberration
ACS	2.1	arccos (in degrees)
APPARENT	10.5	apparent coordinates (CART,POLAR,PRECART)
ASN	2.1	arcsin (in degrees)
ATN	2.1	arctan (in degrees)
ATN2	2.1	arctan (in degrees, correct quadrant)
BRIGHT	7.2	apparent brightness of a planet (SN)
CALDAT	2.2	calender date
CART	2.1	Cartesian coordinates from polar coordinates (CS,SN)
CROSS	11.1	cross-product of two vectors
CS	2.1	cos (in degrees)
CUBR	2.1	cube root
DDD	2.1	decimal degrees from minutes and seconds
DE	5.3	variable order, variable step size, multistep method for numerical integration (STEP,INTRP)
DMS	2.1	minutes and seconds from decimal degrees
DOT	11.1	dot product of two vectors
ECCANOM	4.3	eccentric anomaly (ellipse)
ECLEQU	2.3	transformation ecliptic → equatorial (CS,SN)
ELEMENT	11.1	orbital elements from two positions (ATN2,CROSS,DOT,NORM,POLAR)
ELLIP	4.3	position and velocity in an elliptical orbit (ECCANOM)
EQUECL	2.3	transformation equatorial → ecliptic (CS,SN)
EQUHOR	3.1	transformation equatorial → horizontal (CS,SN,POLAR)
EQUSTD	12.1	standard coordinates from equatorial coordinates (CS,SN)
ETMINUT	9.3	approximation of difference ET-UT
FIND_ETA	11.1	sector-triangle ratio
GAUSVEC	4.5	Gaussian vectors (CS,SN)
GEOCEN	6.5	geocentric coordinates (CS,SN)
HOREQU	3.1	transformation horizontal → equatorial (CS,SN,POLAR)
HYPANOM	4.3	eccentric anomaly (hyperbola)
HYPERB	4.3	position and velocity in a hyperbolic orbit (HYPANOM)
ILLUM	7.2	illumination parameters (ACS)
JUP200	6.3	heliocentric Jupiter coordinates
KEPLER	4.5	position and velocity of a comet (two-body problem for all forms of orbit) (ELLIP,HYPERB,PARAB,ORBECL)
LMST	3.3	local mean sidereal time

LSQFIT	12.3	calculating best fit (least-squares method)
MAR200	A.3	heliocentric Mars coordinates
MER200	A.3	heliocentric Mercury coordinates
MINI_MOON	3.2	reduced-accuracy lunar coordinates
MINI_SUN	3.2	reduced-accuracy solar coordinates
MJD	2.2	Modified Julian Date
MOON	8.2	more accurate lunar ecliptic coordinates
MOONEQU	8.2	more accurate lunar equatorial coordinates (CART, ECLEQU, MOON, NUTEQU, POLAR)
NEP200	A.3	heliocentric Neptune coordinates
NORM	11.1	length of a vector (DOT)
NUTEQU	6.4	nutation in equatorial coordinates
ORBECL	4.5	ecliptic coordinates of a point in the plane of the orbit
ORIENT	7.1	rotational elements of the planets
PARAB	4.4	position and velocity in a parabolic orbit (STUMPPF)
PLU200	A.3	heliocentric Pluto coordinates
PMATECL	2.4	precession matrix (ecliptic coordinates) (CS, SN)
PMATEQU	2.4	precession matrix (equatorial coordinates) (CS, SN)
PN_MATRIX	10.1	matrix for precession and nutation (PMATEQU, NUTEQU)
POLAR	2.1	polar coordinates from Cartesian coordinates (ATN2)
POSANG	7.1	position angle (ATN2)
POSITION	5.2	Cartesian planetary coordinates based on Keplerian orbits (GAUSVEC, ELLIP, ORBECL)
PRECARD	2.4	precession in Cartesian coordinates
QUAD	3.7	quadratic interpolation from three points
ROTATION	7.1	rotation parameters (ATN, ATN2, GAUSVEC)
SAT200	A.3	heliocentric Saturn coordinates
SHAPE	7.1	shape and size of the planets
SITE	10.3	coordinates of a point on the Earth's surface (CS, SN)
SN	2.1	sin (in degrees)
STDEQU	12.1	equatorial coordinates from standard coordinates (CS, SN)
STUMPPF	4.4	Stumpff functions
SUN200	2.5	ecliptic solar coordinates
SUNEQU	9.5	equatorial solar coordin. (CART, ECLEQU, NUTEQU, SUN200)
T_EVAL	8.3	evaluation of a Chebyshev polynomial
T_FIT_LBR	8.3	approximation by Chebyshev polynomials
T_FIT_MOON	8.3	approximation of lunar coordinates by Chebyshev polynomials (MOONEQU, T_FIT_LBR)
T_FIT_SUN	9.5	approximation of solar coordinates by Chebyshev polynomials (SUNEQU, T_FIT_LBR)
TN	2.1	tan (in degrees)
URA200	A.3	heliocentric Uranus coordinates
VEN200	A.3	heliocentric Venus coordinates
XYZKEP	5.4	orbital elements from position and velocity (ATN2)

A.3 Procedures to Calculate Heliocentric Planetary Positions

Mercury

```
(*-----*)
(* MER200: Mercury; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      (      = (JED-2451545.0)/36525      ) *)
(*-----*)
PROCEDURE MER200(T:REAL;VAR L,B,R:REAL);
  CONST P2=6.283185307;
  VAR C1,S1:      ARRAY [-1..9] OF REAL;
      C,S:      ARRAY [-5..0] OF REAL;
      M1,M2,M3,M5,M6: REAL;
      U,V, DL,DR,DB: REAL;
      I:      INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2;
    END;

  PROCEDURE TERM(I1,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
      IF IT=0 THEN ADDTHE(C1[I1],S1[I1],C[I],S[I],U,V)
      ELSE BEGIN U:=U*T; V:=V*T END;
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PROCEDURE PERTVEN; (* Kepler terms and perturbations by Venus *)
    VAR I: INTEGER;
    BEGIN
      C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M2); S[-1]:=-SIN(M2);
      FOR I:=-1 DOWNT0 -4 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
      TERM( 1, 0,0, 259.74,84547.39,-78342.34, 0.01,11683.22,21203.79);
      TERM( 1, 0,1, 2.30, 5.04, -7.52, 0.02, 138.55, -71.01);
      TERM( 1, 0,2, 0.01, -0.01, 0.01, 0.01, -0.19, -0.54);
      TERM( 2, 0,0,-549.71,10394.44, -7955.45, 0.00, 2390.29, 4306.79);
      TERM( 2, 0,1, -4.77, 8.97, -1.53, 0.00, 28.49, -14.18);
      TERM( 2, 0,2, 0.00, 0.00, 0.00, 0.00, -0.04, -0.11);
      TERM( 3, 0,0,-234.04, 1748.74, -1212.86, 0.00, 535.41, 984.33);
      TERM( 3, 0,1, -2.03, 3.48, -0.35, 0.00, 6.56, -2.91);
      TERM( 4, 0,0, -77.64, 332.63, -219.23, 0.00, 124.40, 237.03);
      TERM( 4, 0,1, -0.70, 1.10, -0.08, 0.00, 1.59, -0.59);
      TERM( 5, 0,0, -23.59, 67.28, -43.54, 0.00, 29.44, 58.77);
      TERM( 5, 0,1, -0.23, 0.32, -0.02, 0.00, 0.39, -0.11);
      TERM( 6, 0,0, -6.86, 14.06, -9.18, 0.00, 7.03, 14.84);
```

```

TERM( 6, 0,1, -0.07, 0.09, -0.01, 0.00, 0.10, -0.02);
TERM( 7, 0,0, -1.94, 2.98, -2.02, 0.00, 1.69, 3.80);
TERM( 8, 0,0, -0.54, 0.63, -0.46, 0.00, 0.41, 0.98);
TERM( 9, 0,0, -0.15, 0.13, -0.11, 0.00, 0.10, 0.25);
TERM(-1,-2,0, -0.17, -0.06, -0.05, 0.14, -0.06, -0.07);
TERM( 0,-1,0, 0.24, -0.16, -0.11,-0.16, 0.04, -0.01);
TERM( 0,-2,0, -0.68, -0.25, -0.26, 0.73, -0.16, -0.18);
TERM( 0,-5,0, 0.37, 0.08, 0.06,-0.28, 0.13, 0.12);
TERM( 1,-1,0, 0.58, -0.41, 0.26, 0.36, 0.01, -0.01);
TERM( 1,-2,0, -3.51, -1.23, 0.23,-0.63, -0.05, -0.06);
TERM( 1,-3,0, 0.08, 0.53, -0.11, 0.04, 0.02, -0.09);
TERM( 1,-5,0, 1.44, 0.31, 0.30,-1.39, 0.34, 0.29);
TERM( 2,-1,0, 0.15, -0.11, 0.09, 0.12, 0.02, -0.04);
TERM( 2,-2,0, -1.99, -0.66, 0.65,-1.91, -0.20, 0.03);
TERM( 2,-3,0, -0.34, -1.28, 0.97,-0.26, 0.03, 0.03);
TERM( 2,-4,0, -0.33, 0.35, -0.13,-0.13, -0.01, 0.00);
TERM( 2,-5,0, 7.19, 1.56, -0.05, 0.12, 0.06, 0.05);
TERM( 3,-2,0, -0.52, -0.18, 0.13,-0.39, -0.16, 0.03);
TERM( 3,-3,0, -0.11, -0.42, 0.36,-0.10, -0.05, -0.05);
TERM( 3,-4,0, -0.19, 0.22, -0.23,-0.20, -0.01, 0.02);
TERM( 3,-5,0, 2.77, 0.49, -0.45, 2.56, 0.40, -0.12);
TERM( 4,-5,0, 0.67, 0.12, -0.09, 0.47, 0.24, -0.08);
TERM( 5,-5,0, 0.18, 0.03, -0.02, 0.12, 0.09, -0.03);
END;
```

PROCEDURE PERTTEAR; (* perturbations by the Earth *)

```

VAR I: INTEGER;
BEGIN
  C[-1]:=COS(M3); S[-1]:=-SIN(M3);
  FOR I:=-1 DOWNTO -3 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
  TERM( 0,-4,0, -0.11, -0.07, -0.08, 0.11, -0.02, -0.04);
  TERM( 1,-1,0, 0.10, -0.20, 0.15, 0.07, 0.00, 0.00);
  TERM( 1,-2,0, -0.35, 0.28, -0.13,-0.17, -0.01, 0.00);
  TERM( 1,-4,0, -0.67, -0.45, 0.00, 0.01, -0.01, -0.01);
  TERM( 2,-2,0, -0.20, 0.16, -0.16,-0.20, -0.01, 0.02);
  TERM( 2,-3,0, 0.13, -0.02, 0.02, 0.14, 0.01, 0.00);
  TERM( 2,-4,0, -0.33, -0.18, 0.17,-0.31, -0.04, 0.00);
END;
```

PROCEDURE PERTJUP; (* perturbations by Jupiter *)

```

VAR I: INTEGER;
BEGIN
  C[-1]:=COS(M5); S[-1]:=-SIN(M5);
  FOR I:=-1 DOWNTO -2 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
  TERM(-1,-1,0, -0.08, 0.16, 0.15, 0.08, -0.04, 0.01);
  TERM(-1,-2,0, 0.10, -0.06, -0.07,-0.12, 0.07, -0.01);
  TERM( 0,-1,0, -0.31, 0.48, -0.02, 0.13, -0.03, -0.02);
  TERM( 0,-2,0, 0.42, -0.26, -0.38,-0.50, 0.20, -0.03);
  TERM( 1,-1,0, -0.70, 0.01, -0.02,-0.63, 0.00, 0.03);
  TERM( 1,-2,0, 2.61, -1.97, 1.74, 2.32, 0.01, 0.01);
  TERM( 1,-3,0, 0.32, -0.15, 0.13, 0.28, 0.00, 0.00);
  TERM( 2,-1,0, -0.18, 0.01, 0.00,-0.13, -0.03, 0.03);
  TERM( 2,-2,0, 0.75, -0.56, 0.45, 0.60, 0.08, -0.17);
```

```

    TERM( 3,-2,0, 0.20, -0.15, 0.10, 0.14, 0.04, -0.08);
END;

```

```

PROCEDURE PERTSAT; (* perturbations by Saturn *)

```

```

    BEGIN

```

```

        C[-2]:=COS(2*M6); S[-2]:=-SIN(2*M6);

```

```

        TERM( 1,-2,0, -0.19, 0.33, 0.00, 0.00, 0.00, 0.00);

```

```

    END;

```

```

BEGIN (* MER200 *)

```

```

    DL:=0.0; DR:=0.0; DB:=0.0;

```

```

    M1:=P2*FRAC(0.4855407+415.2014314*T);

```

```

    M2:=P2*FRAC(0.1394222+162.5490444*T);

```

```

    M3:=P2*FRAC(0.9937861+ 99.9978139*T);

```

```

    M5:=P2*FRAC(0.0558417+ 8.4298417*T);

```

```

    M6:=P2*FRAC(0.6823333+ 3.3943333*T);

```

```

    C1[0]:=1.0; S1[0]:=0.0;

```

```

    C1[1]:=COS(M1); S1[1]:=SIN(M1); C1[-1]:=C1[1]; S1[-1]:=-S1[1];

```

```

    FOR I:=2 TO 9 DO ADDTHE(C1[I-1],S1[I-1],C1[1],S1[1],C1[I],S1[I]);

```

```

    PERTVEN; PERTEAR; PERTJUP; PERTSAT;

```

```

    DL := DL + (2.8+3.2*T);

```

```

    L:= 360.0*FRAC(0.2151379 + M1/P2 + ((5601.7+1.1*T)*T+DL)/1296.0E3 );

```

```

    R:= 0.3952829 + 0.0000016*T + DR*1.0E-6;

```

```

    B:= ( -2522.15 + (-30.18 + 0.04*T) * T + DB ) / 3600.0;

```

```

END; (* MER200 *)

```

```

(*-----*)

```

Venus

```

(*-----*)

```

```

(* VEN200: Venus; ecliptic coordinates L,B,R (in deg and AU) *)

```

```

(* equinox of date *)

```

```

(* (T: time in Julian centuries since J2000) *)

```

```

(* ( = (JED-2451545.0)/36525 ) *)

```

```

(*-----*)

```

```

PROCEDURE VEN200(T:REAL;VAR L,B,R:REAL);

```

```

    CONST P2=6.283185307;

```

```

    VAR C2,S2: ARRAY [ 0..8] OF REAL;

```

```

        C,S: ARRAY [-8..0] OF REAL;

```

```

        M1,M2,M3,M4,M5,M6: REAL;

```

```

        U,V, DL,DR,DB: REAL;

```

```

        I: INTEGER;

```

```

FUNCTION FRAC(X:REAL):REAL;

```

```

    BEGIN I:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

```

```
PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
```

```
  BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;
```

```
PROCEDURE TERM(I1,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
```

```
  BEGIN
```

```
    IF IT=0 THEN ADDTHE(C2[I1],S2[I1],C[I],S[I],U,V)
```

```
      ELSE BEGIN U:=U*T; V:=V*T END;
```

```
    DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
```

```
  END;
```

```
PROCEDURE PERTMER; (* perturbations by Mercury *)
```

```
  BEGIN
```

```
    C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M1); S[-1]:=-SIN(M1);
```

```
    ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
```

```
    TERM(1,-1,0, 0.00, 0.00, 0.06, -0.09, 0.01, 0.00);
```

```
    TERM(2,-1,0, 0.25, -0.09, -0.09, -0.27, 0.00, 0.00);
```

```
    TERM(4,-2,0, -0.07, -0.08, -0.14, 0.14, -0.01, -0.01);
```

```
    TERM(5,-2,0, -0.35, 0.08, 0.02, 0.09, 0.00, 0.00);
```

```
  END;
```

```
PROCEDURE PERTEAR; (* Kepler terms and perturbations by the Earth *)
```

```
  VAR I: INTEGER;
```

```
  BEGIN
```

```
    C[-1]:=COS(M3); S[-1]:=-SIN(M3);
```

```
    FOR I:=-1 DOWNT0 -7 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
```

```
    TERM(1, 0,0, 2.37,2793.23,-4899.07, 0.11,9995.27,7027.22);
```

```
    TERM(1, 0,1, 0.10, -19.65, 34.40, 0.22, 64.95, -86.10);
```

```
    TERM(1, 0,2, 0.06, 0.04, -0.07, 0.11, -0.55, -0.07);
```

```
    TERM(2, 0,0,-170.42, 73.13, -16.59, 0.00, 67.71, 47.56);
```

```
    TERM(2, 0,1, 0.93, 2.91, 0.23, 0.00, -0.03, -0.92);
```

```
    TERM(3, 0,0, -2.31, 0.90, -0.08, 0.00, 0.04, 2.09);
```

```
    TERM(1,-1,0, -2.38, -4.27, 3.27, -1.82, 0.00, 0.00);
```

```
    TERM(1,-2,0, 0.09, 0.00, -0.08, 0.05, -0.02, -0.25);
```

```
    TERM(2,-2,0, -9.57, -5.93, 8.57,-13.83, -0.01, -0.01);
```

```
    TERM(2,-3,0, -2.47, -2.40, 0.83, -0.95, 0.16, 0.24);
```

```
    TERM(3,-2,0, -0.09, -0.05, 0.08, -0.13, -0.28, 0.12);
```

```
    TERM(3,-3,0, 7.12, 0.32, -0.62, 13.76, -0.07, 0.01);
```

```
    TERM(3,-4,0, -0.65, -0.17, 0.18, -0.73, 0.10, 0.05);
```

```
    TERM(3,-5,0, -1.08, -0.95, -0.17, 0.22, -0.03, -0.03);
```

```
    TERM(4,-3,0, 0.06, 0.00, -0.01, 0.08, 0.14, -0.18);
```

```
    TERM(4,-4,0, 0.93, -0.46, 1.06, 2.13, -0.01, 0.01);
```

```
    TERM(4,-5,0, -1.53, 0.38, -0.64, -2.54, 0.27, 0.00);
```

```
    TERM(4,-6,0, -0.17, -0.05, 0.03, -0.11, 0.02, 0.00);
```

```
    TERM(5,-5,0, 0.18, -0.28, 0.71, 0.47, -0.02, 0.04);
```

```
    TERM(5,-6,0, 0.15, -0.14, 0.30, 0.31, -0.04, 0.03);
```

```
    TERM(5,-7,0, -0.08, 0.02, -0.03, -0.11, 0.01, 0.00);
```

```
    TERM(5,-8,0, -0.23, 0.00, 0.01, -0.04, 0.00, 0.00);
```

```
    TERM(6,-6,0, 0.01, -0.14, 0.39, 0.04, 0.00, -0.01);
```

```
    TERM(6,-7,0, 0.02, -0.05, 0.12, 0.04, -0.01, 0.01);
```

```
    TERM(6,-8,0, 0.10, -0.10, 0.19, 0.19, -0.02, 0.02);
```

```
    TERM(7,-7,0, -0.03, -0.06, 0.18, -0.08, 0.00, 0.00);
```

```
    TERM(8,-8,0, -0.03, -0.02, 0.06, -0.08, 0.00, 0.00);
```

```
  END;
```

PROCEDURE PERTMAR; (* perturbations by Mars *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M4); S[-1]:=-SIN(M4);

FOR I:=-1 DOWNT0 -2 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(1,-3,0, -0.65, 1.02, -0.04, -0.02, -0.02, 0.00);

TERM(2,-2,0, -0.05, 0.04, -0.09, -0.10, 0.00, 0.00);

TERM(2,-3,0, -0.50, 0.45, -0.79, -0.89, 0.01, 0.03);

END;

PROCEDURE PERTJUP; (* perturbations by Jupiter *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M5); S[-1]:=-SIN(M5);

FOR I:=-1 DOWNT0 -2 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(0,-1,0, -0.05, 1.56, 0.16, 0.04, -0.08, -0.04);

TERM(1,-1,0, -2.62, 1.40, -2.35, -4.40, 0.02, 0.03);

TERM(1,-2,0, -0.47, -0.08, 0.12, -0.76, 0.04, -0.18);

TERM(2,-2,0, -0.73, -0.51, 1.27, -1.82, -0.01, 0.01);

TERM(2,-3,0, -0.14, -0.10, 0.25, -0.34, 0.00, 0.00);

TERM(3,-3,0, -0.01, 0.04, -0.11, -0.02, 0.00, 0.00);

END;

PROCEDURE PERTSAT; (* perturbations by Saturn *)

BEGIN

C[-1]:=COS(M6); S[-1]:=-SIN(M6);

TERM(0,-1,0, 0.00, 0.21, 0.00, 0.00, 0.00, -0.01);

TERM(1,-1,0, -0.11, -0.14, 0.24, -0.20, 0.01, 0.00);

END;

BEGIN (* VEN200 *)

DL:=0.0; DR:=0.0; DB:=0.0;

M1:=P2*FRAC(0.4861431+415.2018375*T);

M2:=P2*FRAC(0.1400197+162.5494552*T);

M3:=P2*FRAC(0.9944153+ 99.9982208*T);

M4:=P2*FRAC(0.0556297+ 53.1674631*T);

M5:=P2*FRAC(0.0567028+ 8.4305083*T);

M6:=P2*FRAC(0.8830539+ 3.3947206*T);

C2[0]:=1.0; S2[0]:=0.0; C2[1]:=COS(M2); S2[1]:=SIN(M2);

FOR I:=2 TO 8 DO ADDTHE(C2[I-1],S2[I-1],C2[1],S2[1],C2[I],S2[I]);

PERTMER; PERTEAR; PERTMAR; PERTJUP; PERTSAT;

DL:=DL + 2.74*SIN(P2*(0.0764+0.4174*T))

+ 0.27*SIN(P2*(0.9201+0.3307*T));

DL:=DL + (1.9+1.8*T);

L:= 360.0*FRAC(0.3654783 + M2/P2 + ((5071.2+1.1*T)*T+DL)/1296.0E3);

R:= 0.7233482 - 0.0000002*T + DR*1.0E-6;

B:= (-67.70 + (0.04 + 0.01*T) * T + DB) / 3600.0;

END; (* VEN200 *)

(*-----*)

Earth

No specific procedure is provided to calculate the heliocentric coordinates of the Earth. These may be determined at any time from the geocentric solar coordinates, which are obtained by SUN200 (see Chap. 2). All that is required is to alter the sign of the Sun's ecliptic latitude, and add 180° to the ecliptic longitude.

Mars

```
(*-----*)
(* MAR200: Mars; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      ( = (JED-2451545.0)/36525 ) *)
(*-----*)
PROCEDURE MAR200(T:REAL;VAR L,B,R:REAL);
  CONST P2=6.283185307;
  VAR C4,S4:      ARRAY [-2..16] OF REAL;
      C,S:        ARRAY [-9.. 0] OF REAL;
      M2,M3,M4,M5,M6: REAL;
      U,V, DL,DR,DB: REAL;
      I:          INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1+C2-S1*S2; S:=S1*C2+C1*S2; END;

  PROCEDURE TERM(I1,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
      IF IT=0 THEN ADDTHE(C4[I1],S4[I1],C[I],S[I],U,V)
        ELSE BEGIN U:=U*T; V:=V*T END;
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PROCEDURE PERTVEN; (* perturbations by Venus *)
    BEGIN
      C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M2); S[-1]:=-SIN(M2);
      ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
      TERM( 0,-1,0, -0.01, -0.03,  0.10, -0.04,  0.00,  0.00);
      TERM( 1,-1,0,  0.06,  0.10, -2.06,  0.75,  0.00,  0.00);
      TERM( 2,-1,0, -0.25, -0.57, -2.58,  1.18,  0.05, -0.04);
      TERM( 2,-2,0,  0.02,  0.02,  0.13, -0.14,  0.00,  0.00);
      TERM( 3,-1,0,  3.41,  5.38,  1.87, -1.15,  0.01, -0.01);
      TERM( 3,-2,0,  0.02,  0.02,  0.11, -0.13,  0.00,  0.00);
      TERM( 4,-1,0,  0.32,  0.49, -1.88,  1.21, -0.07,  0.07);
      TERM( 4,-2,0,  0.03,  0.03,  0.12, -0.14,  0.00,  0.00);
      TERM( 5,-1,0,  0.04,  0.06, -0.17,  0.11, -0.01,  0.01);
```



```

TERM( 5,-2,0, 0.11, 0.09, 0.35, -0.43, -0.01, 0.01);
TERM( 6,-2,0, -0.36, -0.28, -0.20, 0.25, 0.00, 0.00);
TERM( 7,-2,0, -0.03, -0.03, 0.11, -0.13, 0.00, -0.01);
END;

```

PROCEDURE PERTEAR; (* Kepler terms and perturbations by the Earth *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M3); S[-1]:=-SIN(M3);

FOR I:=-1 DOWNT0 -8 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(1, 0, 0, -5.32,38481.97,-141856.04, 0.40,-6321.67,1876.89);

TERM(1, 0, 1, -1.12, 37.98, -138.67, -2.93, 37.28, 117.48);

TERM(1, 0, 2, -0.32, -0.03, 0.12, -1.19, 1.04, -0.40);

TERM(2, 0, 0, 28.28, 2285.80, -6608.37, 0.00, -589.35, 174.81);

TERM(2, 0, 1, 1.64, 3.37, -12.93, 0.00, 2.89, 11.10);

TERM(2, 0, 2, 0.00, 0.00, 0.00, 0.00, 0.10, -0.03);

TERM(3, 0, 0, 5.31, 189.29, -461.81, 0.00, -81.98, 18.53);

TERM(3, 0, 1, 0.31, 0.35, -1.36, 0.00, 0.25, 1.19);

TERM(4, 0, 0, 0.81, 17.96, -38.26, 0.00, -6.88, 2.08);

TERM(4, 0, 1, 0.05, 0.04, -0.15, 0.00, 0.02, 0.14);

TERM(5, 0, 0, 0.11, 1.83, -3.48, 0.00, -0.79, 0.24);

TERM(6, 0, 0, 0.02, 0.20, -0.34, 0.00, -0.09, 0.03);

TERM(-1,-1,0, 0.09, 0.06, 0.14, -0.22, 0.02, -0.02);

TERM(0,-1,0, 0.72, 0.49, 1.55, -2.31, 0.12, -0.10);

TERM(1,-1,0, 7.00, 4.92, 13.93,-20.48, 0.08, -0.13);

TERM(2,-1,0, 13.08, 4.89, -4.53, 10.01, -0.05, 0.13);

TERM(2,-2,0, 0.14, 0.05, -0.48, -2.66, 0.01, 0.14);

TERM(3,-1,0, 1.38, 0.56, -2.00, 4.85, -0.01, 0.19);

TERM(3,-2,0, -6.85, 2.88, 8.38, 21.42, 0.00, 0.03);

TERM(3,-3,0, -0.08, 0.20, 1.20, 0.46, 0.00, 0.00);

TERM(4,-1,0, 0.16, 0.07, -0.19, 0.47, -0.01, 0.05);

TERM(4,-2,0, -4.41, 2.14, -3.33, -7.21, -0.07, -0.09);

TERM(4,-3,0, -0.12, 0.33, 2.22, 0.72, -0.03, -0.02);

TERM(4,-4,0, -0.04, -0.06, -0.36, 0.23, 0.00, 0.00);

TERM(5,-2,0, -0.44, 0.21, -0.70, -1.46, -0.06, -0.07);

TERM(5,-3,0, 0.48, -2.60, -7.25, -1.37, 0.00, 0.00);

TERM(5,-4,0, -0.09, -0.12, -0.66, 0.50, 0.00, 0.00);

TERM(5,-5,0, 0.03, 0.00, 0.01, -0.17, 0.00, 0.00);

TERM(6,-2,0, -0.05, 0.03, -0.07, -0.15, -0.01, -0.01);

TERM(6,-3,0, 0.10, -0.96, 2.36, 0.30, 0.04, 0.00);

TERM(6,-4,0, -0.17, -0.20, -1.09, 0.94, 0.02, -0.02);

TERM(6,-5,0, 0.05, 0.00, 0.00, -0.30, 0.00, 0.00);

TERM(7,-3,0, 0.01, -0.10, 0.32, 0.04, 0.02, 0.00);

TERM(7,-4,0, 0.88, 0.77, 1.86, -2.01, 0.01, -0.01);

TERM(7,-5,0, 0.09, -0.01, -0.05, -0.44, 0.00, 0.00);

TERM(7,-6,0, -0.01, 0.02, 0.10, 0.08, 0.00, 0.00);

TERM(8,-4,0, 0.20, 0.16, -0.53, 0.64, -0.01, 0.02);

TERM(8,-5,0, 0.17, -0.03, -0.14, -0.84, 0.00, 0.01);

TERM(8,-6,0, -0.02, 0.03, 0.16, 0.09, 0.00, 0.00);

TERM(9,-5,0, -0.55, 0.15, 0.30, 1.10, 0.00, 0.00);

TERM(9,-6,0, -0.02, 0.04, 0.20, 0.10, 0.00, 0.00);

TERM(10,-5,0, -0.09, 0.03, -0.10, -0.33, 0.00, -0.01);

TERM(10,-6,0, -0.05, 0.11, 0.48, 0.21, -0.01, 0.00);

```

TERM(11,-8,0, 0.10, -0.35, -0.52, -0.15, 0.00, 0.00);
TERM(11,-7,0, -0.01, -0.02, -0.10, 0.07, 0.00, 0.00);
TERM(12,-6,0, 0.01, -0.04, 0.18, 0.04, 0.01, 0.00);
TERM(12,-7,0, -0.05, -0.07, -0.29, 0.20, 0.01, 0.00);
TERM(13,-7,0, 0.23, 0.27, 0.25, -0.21, 0.00, 0.00);
TERM(14,-7,0, 0.02, 0.03, -0.10, 0.09, 0.00, 0.00);
TERM(14,-8,0, 0.05, 0.01, 0.03, -0.23, 0.00, 0.03);
TERM(15,-8,0, -1.53, 0.27, 0.06, 0.42, 0.00, 0.00);
TERM(16,-8,0, -0.14, 0.02, -0.10, -0.55, -0.01, -0.02);
TERM(16,-9,0, 0.03, -0.06, -0.25, -0.11, 0.00, 0.00);

```

END;

PROCEDURE PERTJUP; (* perturbations by Jupiter *)

VAR I: INTEGER;

BEGIN

```

C[-1]:=COS(M5); S[-1]:=-SIN(M5);
FOR I:=-1 DOWNT0 -4 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
TERM(-2,-1,0, 0.05, 0.03, 0.08, -0.14, 0.01, -0.01);
TERM(-1,-1,0, 0.39, 0.27, 0.92, -1.50, -0.03, -0.06);
TERM(-1,-2,0, -0.16, 0.03, 0.13, 0.67, -0.01, 0.06);
TERM(-1,-3,0, -0.02, 0.01, 0.05, 0.09, 0.00, 0.01);
TERM( 0,-1,0, 3.56, 1.13, -5.41, -7.18, -0.25, -0.24);
TERM( 0,-2,0, -1.44, 0.25, 1.24, 7.96, 0.02, 0.31);
TERM( 0,-3,0, -0.21, 0.11, 0.55, 1.04, 0.01, 0.05);
TERM( 0,-4,0, -0.02, 0.02, 0.11, 0.11, 0.00, 0.01);
TERM( 1,-1,0, 16.67, -19.15, 61.00, 53.36, -0.06, -0.07);
TERM( 1,-2,0, -21.64, 3.18, -7.77, -54.64, -0.31, 0.50);
TERM( 1,-3,0, -2.82, 1.45, -2.53, -5.73, 0.01, 0.07);
TERM( 1,-4,0, -0.31, 0.28, -0.34, -0.51, 0.00, 0.00);
TERM( 2,-1,0, 2.15, -2.29, 7.04, 6.94, 0.33, 0.19);
TERM( 2,-2,0, -15.69, 3.31, -15.70, -73.17, -0.17, -0.25);
TERM( 2,-3,0, -1.73, 1.95, -9.19, -7.20, 0.02, -0.03);
TERM( 2,-4,0, -0.01, 0.33, -1.42, 0.08, 0.01, -0.01);
TERM( 2,-5,0, 0.03, 0.03, -0.13, 0.12, 0.00, 0.00);
TERM( 3,-1,0, 0.26, -0.26, 0.73, 0.71, 0.08, 0.04);
TERM( 3,-2,0, -2.06, 0.46, -1.61, -6.72, -0.13, -0.25);
TERM( 3,-3,0, -1.28, -0.27, 2.21, -6.90, -0.04, -0.02);
TERM( 3,-4,0, -0.22, 0.08, -0.44, -1.25, 0.00, 0.01);
TERM( 3,-5,0, -0.02, 0.03, -0.15, -0.08, 0.00, 0.00);
TERM( 4,-1,0, 0.03, -0.03, 0.08, 0.08, 0.01, 0.01);
TERM( 4,-2,0, -0.26, 0.06, -0.17, -0.70, -0.03, -0.05);
TERM( 4,-3,0, -0.20, -0.05, 0.22, -0.79, -0.01, -0.02);
TERM( 4,-4,0, -0.11, -0.14, 0.93, -0.60, 0.00, 0.00);
TERM( 4,-5,0, -0.04, -0.02, 0.09, -0.23, 0.00, 0.00);
TERM( 5,-4,0, -0.02, -0.03, 0.13, -0.09, 0.00, 0.00);
TERM( 5,-5,0, 0.00, -0.03, 0.21, 0.01, 0.00, 0.00);

```

END;

PROCEDURE PERTSAT; (* perturbations by Saturn *)

VAR I: INTEGER;

BEGIN

```

C[-1]:=COS(M6); S[-1]:=-SIN(M6);
FOR I:=-1 DOWNT0 -3 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

```

```

TERM(-1,-1,0, 0.03, 0.13, 0.48, -0.13, 0.02, 0.00);
TERM( 0,-1,0, 0.27, 0.84, 0.40, -0.43, 0.01, -0.01);
TERM( 0,-2,0, 0.12, -0.04, -0.33, -0.65, -0.01, -0.02);
TERM( 0,-3,0, 0.02, -0.01, -0.07, -0.08, 0.00, 0.00);
TERM( 1,-1,0, 1.12, 0.76, -2.66, 3.91, -0.01, 0.01);
TERM( 1,-2,0, 1.49, -0.95, 3.07, 4.83, 0.04, -0.05);
TERM( 1,-3,0, 0.21, -0.18, 0.55, 0.64, 0.00, 0.00);
TERM( 2,-1,0, 0.12, 0.10, -0.29, 0.34, -0.01, 0.02);
TERM( 2,-2,0, 0.51, -0.36, 1.61, 2.25, 0.03, 0.01);
TERM( 2,-3,0, 0.10, -0.10, 0.50, 0.43, 0.00, 0.00);
TERM( 2,-4,0, 0.01, -0.02, 0.11, 0.05, 0.00, 0.00);
TERM( 3,-2,0, 0.07, -0.05, 0.16, 0.22, 0.01, 0.01);
END;

```

```
BEGIN (* MAR200 *)
```

```

DL:=0.0; DR:=0.0; DB:=0.0;
M2:=P2*FRAC(0.1382208+162.5482542*T);
M3:=P2*FRAC(0.9926208+99.9970236*T);
M4:=P2*FRAC(0.0538553+ 53.1662736*T);
M5:=P2*FRAC(0.0548944+ 8.4290611*T);
M6:=P2*FRAC(0.8811167+ 3.3935250*T);
C4[0]:=1.0; S4[0]:=0.0; C4[1]:=COS(M4); S4[1]:=SIN(M4);
FOR I:=2 TO 16 DO ADDTHE(C4[I-1],S4[I-1],C4[I],S4[I],C4[I],S4[I]);
FOR I:=-2 TO -1 DO BEGIN C4[I]:=C4[-I]; S4[I]:=-S4[-I] END;
PERTVEN; PERTEAR; PERTJUP; PERTSAT;
DL:=DL + 52.49*SIN(P2*(0.1868+0.0549*T))
      + 0.61*SIN(P2*(0.9220+0.3307*T))
      + 0.32*SIN(P2*(0.4731+2.1485*T))
      + 0.28*SIN(P2*(0.9467+0.1133*T));
DL:=DL + (0.14+0.87*T-0.11*T*T);
L:= 360.0*FRAC(0.9334591 + M4/P2 + ((6615.5+1.1*T)*T+DL)/1296.0E3 );
R:= 1.5303352 + 0.0000131*T + DR*1.0E-6;
B:= ( 596.32 + (-2.92 - 0.10*T) * T + DB ) / 3600.0;
END;

```

```
(*-----*)
```

Jupiter

The complete program listing for the JUP200 procedure is given and discussed in Chap. 6.

Saturn

```

(*-----*)
(* SAT200: Saturn; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      (      = (JED-2451545.0)/36525      ) *)
(*-----*)
PROCEDURE SAT200(T:REAL;VAR L,B,R:REAL);
  CONST P2=6.283185307;
  VAR C6,S6:      ARRAY [ 0..11] OF REAL;
      C,S:      ARRAY [-6.. 1] OF REAL;
      M5,M6,M7,M8:  REAL;
      U,V, DL,DR,DB: REAL;
      I:      INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN  X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X  END;

  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN  C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;

  PROCEDURE TERM(I6,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
      IF IT=0 THEN ADDTHE(C6[I6],S6[I6],C[I],S[I],U,V)
        ELSE BEGIN U:=U*T; V:=V*T  END;
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PROCEDURE PERTJUP; (* Kepler terms and perturbations by Jupiter *)
    VAR I: INTEGER;
    BEGIN
      C[0]:=1.0; S[0]:=0.0; C[1]:=COS(M5); S[1]:=SIN(M5);
      FOR I:=0 DOWNT0 -5 DO ADDTHE(C[I],S[I],C[1],-S[1],C[I-1],S[I-1]);
      TERM( 0,-1,0, 12.0, -1.4, -13.9, 6.4, 1.2, -1.8);
      TERM( 0,-2,0, 0.0, -0.2, -0.9, 1.0, 0.0, -0.1);
      TERM( 1, 1,0, 0.9, 0.4, -1.8, 1.9, 0.2, 0.2);
      TERM( 1, 0,0, -348.3,22907.7,-52915.5, -752.2,-3266.5,8314.4);
      TERM( 1, 0,1, -225.2, -146.2, 337.7, -521.3, 79.6, 17.4);
      TERM( 1, 0,2, 1.3, -1.4, 3.2, 2.9, 0.1, -0.4);
      TERM( 1,-1,0, -1.0, -30.7, 108.6, -815.0, -3.6, -9.3);
      TERM( 1,-2,0, -2.0, -2.7, -2.1, -11.9, -0.1, -0.4);
      TERM( 2, 1,0, 0.1, 0.2, -1.0, 0.3, 0.0, 0.0);
      TERM( 2, 0,0, 44.2, 724.0, -1464.3, -34.7, -188.7, 459.1);
      TERM( 2, 0,1, -17.0, -11.3, 18.9, -28.6, 1.0, -3.7);
      TERM( 2,-1,0, -3.5, -426.6, -546.5, -26.5, -1.6, -2.7);
      TERM( 2,-1,1, 3.5, -2.2, -2.6, -4.3, 0.0, 0.0);
      TERM( 2,-2,0, 10.5, -30.9, -130.5, -52.3, -1.9, 0.2);
      TERM( 2,-3,0, -0.2, -0.4, -1.2, -0.1, -0.1, 0.0);
      TERM( 3, 0,0, 6.5, 30.5, -61.1, 0.4, -11.6, 28.1);
      TERM( 3, 0,1, -1.2, -0.7, 1.1, -1.8, -0.2, -0.6);
      TERM( 3,-1,0, 29.0, -40.2, 98.2, 45.3, 3.2, -9.4);
      TERM( 3,-1,1, 0.6, 0.6, -1.0, 1.3, 0.0, 0.0);
    END;

```

```

TERM( 3,-2,0, -27.0, -21.1, -66.5, 8.1, -19.8, 5.4);
TERM( 3,-2,1, 0.9, -0.5, -0.4, -2.0, -0.1, -0.8);
TERM( 3,-3,0, -5.4, -4.1, -19.1, 26.2, -0.1, -0.1);
TERM( 4, 0,0, 0.6, 1.4, -3.0, -0.2, -0.6, 1.6);
TERM( 4,-1,0, 1.5, -2.5, 12.4, 4.7, 1.0, -1.1);
TERM( 4,-2,0, -821.9, -9.6, -26.0, 1873.6, -70.5, -4.4);
TERM( 4,-2,1, 4.1, -21.9, -50.3, -9.9, 0.7, -3.0);
TERM( 4,-3,0, -2.0, -4.7, -19.3, 8.2, -0.1, -0.3);
TERM( 4,-4,0, -1.5, 1.3, 6.5, 7.3, 0.0, 0.0);
TERM( 5,-2,0, -2627.6, -1277.3, 117.4, -344.1, -13.8, -4.3);
TERM( 5,-2,1, 63.0, -98.6, 12.7, 6.7, 0.1, -0.2);
TERM( 5,-2,2, 1.7, 1.2, -0.2, 0.3, 0.0, 0.0);
TERM( 5,-3,0, 0.4, -3.6, -11.3, -1.6, 0.0, -0.3);
TERM( 5,-4,0, -1.4, 0.3, 1.5, 6.3, -0.1, 0.0);
TERM( 5,-5,0, 0.3, 0.6, 3.0, -1.7, 0.0, 0.0);
TERM( 6,-2,0, -146.7, -73.7, 166.4, -334.3, -43.6, -46.7);
TERM( 6,-2,1, 5.2, -6.8, 15.1, 11.4, 1.7, -1.0);
TERM( 6,-3,0, 1.5, -2.9, -2.2, -1.3, 0.1, -0.1);
TERM( 6,-4,0, -0.7, -0.2, -0.7, 2.8, 0.0, 0.0);
TERM( 6,-5,0, 0.0, 0.5, 2.5, -0.1, 0.0, 0.0);
TERM( 6,-6,0, 0.3, -0.1, -0.3, -1.2, 0.0, 0.0);
TERM( 7,-2,0, -9.6, -3.9, 9.6, -18.6, -4.7, -5.3);
TERM( 7,-2,1, 0.4, -0.5, 1.0, 0.9, 0.3, -0.1);
TERM( 7,-3,0, 3.0, 5.3, 7.5, -3.5, 0.0, 0.0);
TERM( 7,-4,0, 0.2, 0.4, 1.6, -1.3, 0.0, 0.0);
TERM( 7,-5,0, -0.1, 0.2, 1.0, 0.5, 0.0, 0.0);
TERM( 7,-6,0, 0.2, 0.0, 0.2, -1.0, 0.0, 0.0);
TERM( 8,-2,0, -0.7, -0.2, 0.6, -1.2, -0.4, -0.4);
TERM( 8,-3,0, 0.5, 1.0, -2.0, 1.5, 0.1, 0.2);
TERM( 8,-4,0, 0.4, 1.3, 3.6, -0.9, 0.0, -0.1);
TERM( 9,-4,0, 4.0, -8.7, -19.9, -9.9, 0.2, -0.4);
TERM( 9,-4,1, 0.5, 0.3, 0.8, -1.8, 0.0, 0.0);
TERM(10,-4,0, 21.3, -16.8, 3.3, 3.3, 0.2, -0.2);
TERM(10,-4,1, 1.0, 1.7, -0.4, 0.4, 0.0, 0.0);
TERM(11,-4,0, 1.6, -1.3, 3.0, 3.7, 0.8, -0.2);
END;

```

PROCEDURE PERTURA; (* perturbations by Uranus *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M7); S[-1]:=-SIN(M7);

FOR I:=-1 DOWNT0 -4 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(0,-1,0, 1.0, 0.7, 0.4, -1.5, 0.1, 0.0);

TERM(0,-2,0, 0.0, -0.4, -1.1, 0.1, -0.1, -0.1);

TERM(0,-3,0, -0.9, -1.2, -2.7, 2.1, -0.5, -0.3);

TERM(1,-1,0, 7.8, -1.5, 2.3, 12.7, 0.0, 0.0);

TERM(1,-2,0, -1.1, -8.1, 5.2, -0.3, -0.3, -0.3);

TERM(1,-3,0, -16.4, -21.0, -2.1, 0.0, 0.4, 0.0);

TERM(2,-1,0, 0.6, -0.1, 0.1, 1.2, 0.1, 0.0);

TERM(2,-2,0, -4.9, -11.7, 31.5, -13.3, 0.0, -0.2);

TERM(2,-3,0, 19.1, 10.0, -22.1, 42.1, 0.1, -1.1);

TERM(2,-4,0, 0.9, -0.1, 0.1, 1.4, 0.0, 0.0);

TERM(3,-2,0, -0.4, -0.9, 1.7, -0.8, 0.0, -0.3);

```

      TERM( 3,-3,0,    2.3,    0.0,    1.0,    5.7,    0.3,    0.3);
      TERM( 3,-4,0,    0.3,   -0.7,    2.0,    0.7,    0.0,    0.0);
      TERM( 3,-5,0,   -0.1,   -0.4,    1.1,   -0.3,    0.0,    0.0);
END;

```

PROCEDURE PERTNEP; (* perturbations by Neptune *)

```

BEGIN
  C[-1]:=COS(M8); S[-1]:=-SIN(M8);
  ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
  TERM( 1,-1,0,   -1.3,   -1.2,    2.3,   -2.5,    0.0,    0.0);
  TERM( 1,-2,0,    1.0,   -0.1,    0.1,    1.4,    0.0,    0.0);
  TERM( 2,-2,0,    1.1,   -0.1,    0.2,    3.3,    0.0,    0.0);
END;

```

PROCEDURE PERTJUR; (* perturbations by Jupiter and Uranus *)

```

  VAR PHI,X,Y: REAL;
  BEGIN
    PHI:=(-2*M5+5*M6-3*M7); X:=COS(PHI); Y:=SIN(PHI);
    DL:=DL-0.8*X-0.1*Y; DR:=DR-0.2*X+1.8*Y; DB:=DB+0.3*X+0.5*Y;
    ADDTHE(X,Y,C6[1],S6[1],X,Y);
    DL:=DL+(+2.4-0.7*T)*X+(27.8-0.4*T)*Y; DR:=DR+2.1*X-0.2*Y;
    ADDTHE(X,Y,C6[1],S6[1],X,Y);
    DL:=DL+0.1*X+1.6*Y; DR:=DR-3.6*X+0.3*Y; DB:=DB-0.2*X+0.6*Y;
  END;

```

BEGIN (* SAT200 *)

```

  DL:=0.0; DR:=0.0; DB:=0.0;
  M5:=P2*FRAC(0.0565314+8.4302963*T); M6:=P2*FRAC(0.6629867+3.3947668*T);
  M7:=P2*FRAC(0.3969537+1.1902586*T); M8:=P2*FRAC(0.7208473+0.6068623*T);
  C6[0]:=1.0; S6[0]:=0.0; C6[1]:=COS(M6); S6[1]:=SIN(M6);
  FOR I:=2 TO 11 DO ADDTHE(C6[I-1],S6[I-1],C6[1],S6[1],C6[I],S6[I]);
  PERTJUP; PERTURA; PERTNEP; PERTJUR;
  L:= 360.0*FRAC(0.2561136 + M6/P2 + ((5018.6+T*1.9)*T +DL)/1296.0E3 );
  R:= 9.557584 - 0.000186*T + DR*1.0E-5;
  B:= ( 175.1 - 10.2*T + DB ) / 3600.0;

```

END; (* SAT200 *)

(*-----*)

Uranus

```

(*-----*)
(* URA200: Uranus; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      (      = (JED-2451545.0)/36525      ) *)
(*-----*)
PROCEDURE URA200(T:REAL;VAR L,B,R:REAL);
  CONST P2=6.283165307;
  VAR C7,S7:      ARRAY [-2..7] OF REAL;
      C,S:      ARRAY [-6..0] OF REAL;
      M5,M6,M7,M8: REAL;
      U,V, DL,DR,DB: REAL;
      I:      INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;

  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;

  PROCEDURE TERM(I7,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
      IF IT=0 THEN ADDTHE(C7[I7],S7[I7],C[I],S[I],U,V)
        ELSE BEGIN U:=U*T; V:=V*T END;
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PROCEDURE PERTJUP; (* perturbations by Jupiter *)
    BEGIN
      C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M5); S[-1]:=-SIN(M5);
      ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
      TERM(-1,-1,0, 0.0, 0.0, -0.1, 1.7, -0.1, 0.0);
      TERM( 0,-1,0, 0.5, -1.2, 18.9, 9.1, -0.9, 0.1);
      TERM( 1,-1,0,-21.2, 48.7, -455.5,-198.8, 0.0, 0.0);
      TERM( 1,-2,0, -0.5, 1.2, -10.9, -4.8, 0.0, 0.0);
      TERM( 2,-1,0, -1.3, 3.2, -23.2, -11.1, 0.3, 0.1);
      TERM( 2,-2,0, -0.2, 0.2, 1.1, 1.5, 0.0, 0.0);
      TERM( 3,-1,0, 0.0, 0.2, -1.8, 0.4, 0.0, 0.0);
    END;

  PROCEDURE PERTSAT; (* perturbations by Saturn *)
    VAR I: INTEGER;
    BEGIN
      C[-1]:=COS(M6); S[-1]:=-SIN(M6);
      FOR I:=-1 DOWNT0 -3 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
      TERM( 0,-1,0, 1.4, -0.5, -6.4, 9.0, -0.4, -0.8);
      TERM( 1,-1,0,-16.6, -12.6, 36.7,-336.8, 1.0, 0.3);
      TERM( 1,-2,0, -0.7, -0.3, 0.5, -7.5, 0.1, 0.0);
      TERM( 2,-1,0, 20.0, -141.6, -587.1,-107.0, 3.1, -0.8);
      TERM( 2,-1,1, 1.0, 1.4, 5.8, -4.0, 0.0, 0.0);
      TERM( 2,-2,0, 1.6, -3.8, -35.6, -16.0, 0.0, 0.0);
    END;

```

```

TERM( 3,-1,0, 75.3, -100.9, 128.9, 77.5, -0.8, 0.1);
TERM( 3,-1,1, 0.2, 1.6, -1.9, 0.3, 0.0, 0.0);
TERM( 3,-2,0, 2.3, -1.3, -9.5, -17.9, 0.0, 0.1);
TERM( 3,-3,0, -0.7, -0.5, -4.9, 6.8, 0.0, 0.0);
TERM( 4,-1,0, 3.4, -5.0, 21.6, 14.3, -0.8, -0.5);
TERM( 4,-2,0, 1.9, 0.1, 1.2, -12.1, 0.0, 0.0);
TERM( 4,-3,0, -0.1, -0.4, -3.9, 1.2, 0.0, 0.0);
TERM( 4,-4,0, -0.2, 0.1, 1.6, 1.8, 0.0, 0.0);
TERM( 5,-1,0, 0.2, -0.3, 1.0, 0.6, -0.1, 0.0);
TERM( 5,-2,0, -2.2, -2.2, -7.7, 8.5, 0.0, 0.0);
TERM( 5,-3,0, 0.1, -0.2, -1.4, -0.4, 0.0, 0.0);
TERM( 5,-4,0, -0.1, 0.0, 0.1, 1.2, 0.0, 0.0);
TERM( 6,-2,0, -0.2, -0.6, 1.4, -0.7, 0.0, 0.0);
END;

```

PROCEDURE PERTNEP; (* Kepler terms and perturbations by Neptune *)

VAR I: INTEGER;

BEGIN

C[-1]:=COS(M8); S[-1]:=-SIN(M8);

FOR I:=-1 DOWNT0 -7 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);

TERM(1, 0,0,-78.1,19518.1,-90718.2,-334.7,2759.5,-311.9);

TERM(1, 0,1,-81.6, 107.7, -497.4,-379.5, -2.8,-43.7);

TERM(1, 0,2, -6.6, -3.1, 14.4, -30.6, -0.4, -0.5);

TERM(1, 0,3, 0.0, -0.5, 2.4, 0.0, 0.0, 0.0);

TERM(2, 0,0, -2.4, 586.1, -2145.2, -15.3, 130.6, -14.3);

TERM(2, 0,1, -4.5, 6.6, -24.2, -17.8, 0.7, -1.6);

TERM(2, 0,2, -0.4, 0.0, 0.1, -1.4, 0.0, 0.0);

TERM(3, 0,0, 0.0, 24.5, -76.2, -0.6, 7.0, -0.7);

TERM(3, 0,1, -0.2, 0.4, -1.4, -0.8, 0.1, -0.1);

TERM(4, 0,0, 0.0, 1.1, -3.0, 0.1, 0.4, 0.0);

TERM(-1,-1,0, -0.2, 0.2, 0.7, 0.7, -0.1, 0.0);

TERM(0,-1,0, -2.6, 2.5, 8.7, 10.5, -0.4, -0.1);

TERM(1,-1,0,-28.4, 20.3, -51.4, -72.0, 0.0, 0.0);

TERM(1,-2,0, -0.6, -0.1, 4.2, -14.6, 0.2, 0.4);

TERM(1,-3,0, 0.2, 0.5, 3.4, -1.6, -0.1, 0.1);

TERM(2,-1,0, -1.8, 1.3, -5.5, -7.7, 0.0, 0.3);

TERM(2,-2,0, 29.4, 10.2, -29.0, 83.2, 0.0, 0.0);

TERM(2,-3,0, 8.8, 17.8, -41.9, 21.5, -0.1, -0.3);

TERM(2,-4,0, 0.0, 0.1, -2.1, -0.9, 0.1, 0.0);

TERM(3,-2,0, 1.5, 0.5, -1.7, 5.1, 0.1, -0.2);

TERM(3,-3,0, 4.4, 14.6, -64.3, 25.2, 0.1, -0.1);

TERM(3,-4,0, 2.4, -4.5, 12.0, 6.2, 0.0, 0.0);

TERM(3,-5,0, 2.9, -0.9, 2.1, 6.2, 0.0, 0.0);

TERM(4,-3,0, 0.3, 1.0, -4.0, 1.1, 0.1, -0.1);

TERM(4,-4,0, 2.1, -2.7, 17.9, 14.0, 0.0, 0.0);

TERM(4,-5,0, 3.0, -0.4, 2.3, 17.6, -0.1, -0.1);

TERM(4,-6,0, -0.6, -0.5, 1.1, -1.6, 0.0, 0.0);

TERM(5,-4,0, 0.2, -0.2, 1.0, 0.8, 0.0, 0.0);

TERM(5,-5,0, -0.9, -0.1, 0.6, -7.1, 0.0, 0.0);

TERM(5,-6,0, -0.5, -0.6, 3.8, -3.6, 0.0, 0.0);

TERM(5,-7,0, 0.0, -0.5, 3.0, 0.1, 0.0, 0.0);

TERM(6,-6,0, 0.2, 0.3, -2.7, 1.6, 0.0, 0.0);

TERM(6,-7,0, -0.1, 0.2, -2.0, -0.4, 0.0, 0.0);


```

TERM( 7,-7,0, 0.1, -0.2, 1.3, 0.5, 0.0, 0.0);
TERM( 7,-8,0, 0.1, 0.0, 0.4, 0.9, 0.0, 0.0);
END;

```

PROCEDURE PERTJSU; (* perturbations by Jupiter and Saturn *)

VAR I: INTEGER;

BEGIN

```

C[-1]:=COS(M6);          S[-1]:=-SIN(M6);
C[-4]:=COS(-4*M6+2*M5); S[-4]:= SIN(-4*M6+2*M5);
FOR I:=-4 DOWNT0 -5 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
TERM(-2,-4,0, -0.7, 0.4, -1.5, -2.5, 0.0, 0.0);
TERM(-1,-4,0, -0.1, -0.1, -2.2, 1.0, 0.0, 0.0);
TERM( 1,-5,0, 0.1, -0.4, 1.4, 0.2, 0.0, 0.0);
TERM( 1,-6,0, 0.4, 0.5, -0.8, -0.8, 0.0, 0.0);
TERM( 2,-6,0, 5.7, 6.3, 28.5, -25.5, 0.0, 0.0);
TERM( 2,-6,1, 0.1, -0.2, -1.1, -0.6, 0.0, 0.0);
TERM( 3,-6,0, -1.4, 29.2, -11.4, 1.1, 0.0, 0.0);
TERM( 3,-6,1, 0.8, -0.4, 0.2, 0.3, 0.0, 0.0);
TERM( 4,-6,0, 0.0, 1.3, -6.0, -0.1, 0.0, 0.0);
END;

```

BEGIN (* URA200 *)

```

DL:=0.0; DR:=0.0; DB:=0.0;
M5:=P2*FRAC(0.0564472+8.4302889*T); M6:=P2*FRAC(0.8829611+3.3947583*T);
M7:=P2*FRAC(0.3967117+1.1902849*T); M8:=P2*FRAC(0.7216833+0.6068528*T);
C7[0]:=1.0; S7[0]:=0.0; C7[1]:=COS(M7); S7[1]:=SIN(M7);
FOR I:=2 TO 7 DO ADDTHE(C7[I-1],S7[I-1],C7[1],S7[1],C7[I],S7[I]);
FOR I:=1 TO 2 DO BEGIN C7[-I]:=C7[I]; S7[-I]:=-S7[I]; END;
PERTJUP; PERTSAT; PERTNEP; PERTJSU;
L:= 360.0*FRAC(0.4734843 + M7/P2 + ((5082.3+34.2*T)*T+DL)/1296.0E3 );
R:= 19.211991 + (-0.000333-0.000005*T)*T + DR*1.0E-5;
B:= (-130.61 + (-0.54+0.04*T)*T + DB ) / 3600.0;

```

END; (* URA200 *)

(*-----*)

Neptune

```

(*-----*)
(* NEP200: Neptune; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date *)
(*      (T: time in Julian centuries since J2000) *)
(*      (      = (JED-2451545.0)/36525      ) *)
(*-----*)
PROCEDURE NEP200(T:REAL;VAR L,B,R:REAL);
  CONST P2=6.283185307;
  VAR C8,S8:      ARRAY [ 0..6] OF REAL;
      C,S:      ARRAY [-6..0] OF REAL;
      M5,M6,M7,M8:  REAL;
      U,V, DL,DR,DB: REAL;
      I: INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN I:=X-TRUNC(X); IF (X<0) THEN I:=X+1.0; FRAC:=I END;

  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1*C2-S1*S2; S:=S1*C2+C1*S2; END;

  PROCEDURE TERM(I1,I,IT:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    BEGIN
      IF IT=0 THEN ADDTHE(C8[I1],S8[I1],C[I],S[I],U,V)
        ELSE BEGIN U:=U*T; V:=V*T END;
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PRDCEURE PERTJUP; (* perturbations by Jupiter *)
  BEGIN
    C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M5); S[-1]:=-SIN(M5);
    ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
    TERM(0,-1,0, 0.1, 0.1, -3.0, 1.8, -0.3, -0.3);
    TERM(1, 0,0, 0.0, 0.0, -15.9, 9.0, 0.0, 0.0);
    TERM(1,-1,0,-17.6, -29.3, 416.1,-250.0, 0.0, 0.0);
    TERM(1,-2,0, -0.4, -0.7, 10.4, -6.2, 0.0, 0.0);
    TERM(2,-1,0, -0.2, -0.4, 2.4, -1.4, 0.4, -0.3);
  END;

  PROCEDURE PERTSAT; (* perturbations by Saturn *)
  BEGIN
    C[0]:=1.0; S[0]:=0.0; C[-1]:=COS(M6); S[-1]:=-SIN(M6);
    ADDTHE(C[-1],S[-1],C[-1],S[-1],C[-2],S[-2]);
    TERM(0,-1,0, -0.1, 0.0, 0.2, -1.8, -0.1, -0.5);
    TERM(1, 0,0, 0.0, 0.0, -8.3, -10.4, 0.0, 0.0);
    TERM(1,-1,0, 13.6, -12.7, 187.5, 201.1, 0.0, 0.0);
    TERM(1,-2,0, 0.4, -0.4, 4.5, 4.5, 0.0, 0.0);
    TERM(2,-1,0, 0.4, -0.1, 1.7, -3.2, 0.2, 0.2);
    TERM(2,-2,0, -0.1, 0.0, -0.2, 2.7, 0.0, 0.0);
  END;

  PROCEDURE PERTURA; (* Kepler terms and perturbations by Uranus *)

```

VAR I: INTEGER;

BEGIN

```

C[0]:=1.0; S[0]:=0.0; C[-1]:=CDS(M7); S[-1]:=-SIN(M7);
FOR I:=-1 DOWNTD -5 DO ADDTHE(C[I],S[I],C[-1],S[-1],C[I-1],S[I-1]);
TERM(1, 0,0, 32.3,3549.5,-25880.2, 235.8,-6360.5,374.0);
TERM(1, 0,1, 31.2, 34.4, -251.4, 227.4, 34.9, 29.3);
TERM(1, 0,2, -1.4, 3.9, -28.6,-10.1, 0.0,-0.9);
TERM(2, 0,0, 6.1, 68.0, -111.4, 2.0, -54.7, 3.7);
TERM(2, 0,1, 0.8, -0.2, -2.1, 2.0, -0.2, 0.8);
TERM(3, 0,0, 0.1, 1.0, -0.7, 0.0, -0.8, 0.1);
TERM(0,-1,0, -0.1, -0.3, -3.6, 0.0, 0.0, 0.0);
TERM(1, 0,0, 0.0, 0.0, 5.5, -6.9, 0.1, 0.0);
TERM(1,-1,0, -2.2, -1.6, -116.3, 163.6, 0.0, -0.1);
TERM(1,-2,0, 0.2, 0.1, -1.2, 0.4, 0.0, -0.1);
TERM(2,-1,0, 4.2, -1.1, -4.4, -34.6, -0.2, 0.1);
TERM(2,-2,0, 8.6, -2.9, -33.4, -97.0, 0.2, 0.1);
TERM(3,-1,0, 0.1, -0.2, 2.1, -1.2, 0.0, 0.1);
TERM(3,-2,0, -4.6, 9.3, 38.2, 19.8, 0.1, 0.1);
TERM(3,-3,0, -0.5, 1.7, 23.5, 7.0, 0.0, 0.0);
TERM(4,-2,0, 0.2, 0.8, 3.3, -1.5, -0.2, -0.1);
TERM(4,-3,0, 0.9, 1.7, 17.9, -9.1, -0.1, 0.0);
TERM(4,-4,0, -0.4, -0.4, -6.2, 4.8, 0.0, 0.0);
TERM(5,-3,0, -1.6, -0.5, -2.2, 7.0, 0.0, 0.0);
TERM(5,-4,0, -0.4, -0.1, -0.7, 5.5, 0.0, 0.0);
TERM(5,-5,0, 0.2, 0.0, 0.0, -3.5, 0.0, 0.0);
TERM(6,-4,0, -0.3, 0.2, 2.1, 2.7, 0.0, 0.0);
TERM(6,-5,0, 0.1, -0.1, -1.4, -1.4, 0.0, 0.0);
TERM(6,-6,0, -0.1, 0.1, 1.4, 0.7, 0.0, 0.0);

```

END;

BEGIN (* NEP200 *)

```

DL:=0.0; DR:=0.0; DB:=0.0;
M5:=P2*FRAC(0.0563867+8.4298907*T); M6:=P2*FRAC(0.8825086+3.3957748*T);
M7:=P2*FRAC(0.3965358+1.1902851*T); M8:=P2*FRAC(0.7214906+0.6068526*T);
C8[0]:=1.0; S8[0]:=0.0; C8[1]:=COS(M8); S8[1]:=SIN(M8);
FOR I:=2 TO 6 DO ADDTHE(C8[I-1],S8[I-1],C8[1],S8[1],C8[I],S8[I]);
PERTJUP; PERTSAT; PERTURA;
L:= 360.0*FRAC(0.1254046 + M8/P2 + ((4982.8-21.3*T)*T+DL)/1296.0E3 );
R:= 30.072984 + (0.001234+0.000003*T) * T + DR*1.0E-5;
B:= ( 54.77 + ( 0.26 + 0.06*T) * T + DB ) / 3600.0;

```

END; (* NEP200 *)

(*-----*)

Pluto

The structure of the PLU200 routine differs from the other procedures. The coordinates are first calculated relative to the fixed ecliptic of 1950, and then transformed to the equinox of date. This method is necessary because of the high inclination of Pluto's orbit. Complete expansion of the coordinates would lead to numerous secular terms. It should also be noted that PLU200 can only be used for calculations relating to the years between 1890 and 2100. The reason for this is that the series expansion used was not derived from perturbation theory, but from a Fourier analysis of a numerically integrated ephemeris covering that period of time. Even a few years before 1890 or after 2100, the errors in the calculated coordinates grow very sharply, reaching values of more than 0°5.

```
(*-----*)
(* PLU200: Pluto; ecliptic coordinates L,B,R (in deg and AU) *)
(*      equinox of date; only valid between 1890 and 2100!! *)
(*      (T: time in Julian centuries since J2000) *)
(*      (   = (JED-2451545.0)/36525 ) *)
(*-----*)
PROCEDURE PLU200(T:REAL;VAR L,B,R:REAL);

  CONST P2=6.283185307;
  VAR C9,S9:   ARRAY [ 0..6] OF REAL;
      C,S:     ARRAY [-3..2] OF REAL;
      M5,M6,M9: REAL;
      DL,DR,DB: REAL;
      I:       INTEGER;

  FUNCTION FRAC(X:REAL):REAL;
    BEGIN X:=X-TRUNC(X); IF (X<0) THEN X:=X+1.0; FRAC:=X END;
  PROCEDURE ADDTHE(C1,S1,C2,S2:REAL; VAR C,S:REAL);
    BEGIN C:=C1+C2-S1*S2; S:=S1*C2+C1*S2; END;

  PRDCEURE TERM(I9,I:INTEGER;DLC,DLS,DRC,DRS,DBC,DBS:REAL);
    VAR U,V: REAL;
    BEGIN
      ADDTHE(C9[I9],S9[I9],C[I],S[I],U,V);
      DL:=DL+DLC*U+DLS*V; DR:=DR+DRC*U+DRS*V; DB:=DB+DBC*U+DBS*V;
    END;

  PRDCEURE PERTJUP; (* Kepler terms and perturbations by Jupiter *)
    VAR I: INTEGER;
    BEGIN
      C[0]:=1.0; S[0]:=0.0; C[1]:=CDS(M5); S[1]:=SIN(M5);
      FOR I:=0 DOWNTO -1 DO ADDTHE(C[I],S[I],C[1],-S[1],C[I-1],S[I-1]);
      ADDTHE(C[1],S[1],C[1],S[1],C[2],S[2]);
      TERM(1, 0, 0.06,100924.08,-960396.0,15965.1,51987.68,-24288.76);
      TERM(2, 0,3274.74, 17835.12,-118252.2, 3632.4,12687.49, -6049.72);
      TERM(3, 0,1543.52, 4631.99, -21446.6, 1167.0, 3504.00, -1853.10);
      TERM(4, 0, 688.99, 1227.08, -4823.4, 213.5, 1048.19, -648.26);
```

```

TERM(5, 0, 242.27, 415.93, -1075.4, 140.6, 302.33, -209.76);
TERM(6, 0, 138.41, 110.91, -308.8, -55.3, 109.52, -93.82);
TERM(3,-1, -0.99, 5.06, -25.6, 19.8, 1.26, -1.96);
TERM(2,-1, 7.15, 5.61, -96.7, 57.2, 1.64, -2.16);
TERM(1,-1, 10.79, 23.13, -390.4, 236.4, -0.33, 0.86);
TERM(0, 1, -0.23, 4.43, 102.8, 63.2, 3.15, 0.34);
TERM(1, 1, -1.10, -0.92, 11.8, -2.3, 0.43, 0.14);
TERM(2, 1, 0.62, 0.84, 2.3, 0.7, 0.05, -0.04);
TERM(3, 1, -0.38, -0.45, 1.2, -0.8, 0.04, 0.05);
TERM(4, 1, 0.17, 0.25, 0.0, 0.2, -0.01, -0.01);
TERM(3,-2, 0.06, 0.07, -0.6, 0.3, 0.03, -0.03);
TERM(2,-2, 0.13, 0.20, -2.2, 1.5, 0.03, -0.07);
TERM(1,-2, 0.32, 0.49, -9.4, 5.7, -0.01, 0.03);
TERM(0,-2, -0.04, -0.07, 2.6, -1.5, 0.07, -0.02);
END;

```

PROCEDURE PERTSAT; (* perturbations by Saturn *)

VAR I: INTEGER;

BEGIN

C[1]:=COS(M6); S[1]:=SIN(M6);

FOR I:=0 DOWNT0 -1 DO ADDTHE(C[I],S[I],C[1],-S[1],C[I-1],S[I-1]);

TERM(1,-1, -29.47, 75.97, -106.4, -204.9, -40.71, -17.55);

TERM(0, 1, -13.88, 18.20, 42.6, -46.1, 1.13, 0.43);

TERM(1, 1, 5.81, -23.48, 15.0, -6.8, -7.48, 3.07);

TERM(2, 1, -10.27, 14.16, -7.9, 0.4, 2.43, -0.09);

TERM(3, 1, 6.86, -10.66, 7.3, -0.3, -2.25, 0.69);

TERM(2,-2, 4.32, 2.00, 0.0, -2.2, -0.24, 0.12);

TERM(1,-2, -5.04, -0.83, -9.2, -3.1, 0.79, -0.24);

TERM(0,-2, 4.25, 2.48, -5.9, -3.3, 0.58, 0.02);

END;

PROCEDURE PERTJUS; (* perturbations by Jupiter and Saturn *)

VAR PHI,X,Y: REAL;

BEGIN

PHI:=(M5-M6); X:=COS(PHI); Y:=SIN(PHI);

DL:=DL-9.11*X+0.12*Y; DR:=DR-3.4*X-3.3*Y; DB:=DB+0.81*X+0.78*Y;

ADDTHE(X,Y,C0[1],S0[1],X,Y);

DL:=DL+5.92*X+0.25*Y; DR:=DR+2.3*X-3.8*Y; DB:=DB-0.67*X-0.51*Y;

END;

PROCEDURE PREC(T:REAL;VAR L,B:REAL); (* process. 1950->equinox of date *)

CONST DEG=57.2957795;

VAR D,PPI,PI,P,C1,S1,C2,S2,C3,S3,X,Y,Z: REAL;

BEGIN

D:=T+0.5; L:=L/DEG; B:=B/DEG;

PPI:=3.044; PI:=2.28E-4*D; P:=(0.0243764+5.39E-6*D)*D;

C1:=COS(PI); C2:=COS(B); C3:=CDS(PPI-L);

S1:=SIN(PI); S2:=SIN(B); S3:=SIN(PPI-L);

X:=C2*C3; Y:=C1*C2*S3-S1*S2; Z:=S1*C2*S3+C1*S2;

B := DEG * ARCTAN(Z / SQRT((1.0-Z)*(1.0+Z)));

IF (X>0) THEN L:=360.0*FRAC((PPI+P-ARCTAN(Y/X))/P2)

ELSE L:=360.0*FRAC((PPI+P-ARCTAN(Y/X))/P2+0.5);

END;

BEGIN (* PLU200 *)

```
DL:=0.0; DR:=0.0; DB:=0.0;
M5:=P2*FRAC(0.0565314+8.4302963*T); M6:=P2*FRAC(0.8829867+3.3947688*T);
M9:=P2*FRAC(0.0385795+0.4026667*T);
C9[0]:=1.0; S9[0]:=0.0; C9[1]:=COS(M9); S9[1]:=SIN(M9);
FOR I:=2 TD 6 DO ADDTHE(C9[I-1],S9[I-1],C9[1],S9[1],C9[I],S9[I]);
PERTJUP; PERTSAT; PERTJUS;
L:= 360.0*FRAC( 0.6232469 + M9/P2 + DL/1296.0E3 );
R:= 40.7247248 + DR * 1.0E-5;
B:= -3.909434 + DB / 3600.0;
PREC(T,L,B);
```

END; (* PLU200 *)

(*-----*)

Symbols

<i>a</i>	semi-major axis
<i>a</i>	station coefficient in longitude
<i>A</i>	azimuth
<i>b</i>	ecliptic latitude
<i>b</i>	station coefficient in latitude
<i>ḃ</i>	change in ecliptic latitude with time
<i>B</i>	ecliptic latitude of the Sun
<i>c</i>	velocity of light ($c = 299\,792\,458$ m/s)
$c_i(E)$	Stumpff functions
<i>C</i>	parallactic angle
<i>d</i>	diameter of the umbra on the Earth
<i>d</i>	rotation axis unit vector
<i>db</i>	perturbation of ecliptic latitude
<i>dl</i>	perturbation of ecliptic longitude
<i>dr</i>	perturbation of distance
<i>D</i>	mean elongation of the Moon from the Sun
<i>D</i>	diameter of the penumbra on the Earth
<i>e</i>	eccentricity
<i>e</i>	vector of unit length
<i>E</i>	eccentric anomaly
<i>E</i>	elongation
ET	Ephemeris Time
<i>f</i>	flattening of the Earth or a planet
<i>f</i>	half apex angle of umbral and penumbral cone
<i>f</i>	coordinate difference in the fundamental plane
<i>F</i>	general: gravitational attraction between two bodies
<i>F</i>	mean distance of the Moon from the ascending node of its orbit
<i>F</i>	focal length
<i>g</i>	coordinate difference in the fundamental plane
<i>G</i>	gravitational constant ($G = 2.959122083 \cdot 10^{-4} \text{AU}^3 M_{\odot}^{-1} \text{d}^{-2}$)
GMST	Greenwich Mean Sidereal Time
<i>h</i>	altitude
<i>h</i>	step size
<i>H</i>	eccentric anomaly for a hyperbolic orbit

<i>i</i>	orbital inclination to the ecliptic
<i>i</i>	phase angle
<i>i</i>	unit vector in the fundamental plane (<i>x</i> -axis)
<i>j</i>	unit vector in the fundamental plane (<i>y</i> -axis)
JD	Julian Date
<i>k</i>	Gaussian gravitational constant ($k = 0.01720209895$)
<i>k</i>	phase
<i>k</i>	ratio of lunar radius to radius of the Earth ($k \approx 0.2725$)
<i>k</i>	unit vector perpendicular to the fundamental plane (<i>z</i> -axis)
<i>l</i>	ecliptic longitude
<i>l</i>	shadow radius in the fundamental plane
<i>l</i>	mean anomaly of the Moon
<i>l</i>	distance between apex of umbral cone and centre of the Earth
<i>l'</i>	mean anomaly of the Sun
<i>l</i>	change in ecliptic longitude with time
<i>L</i>	ecliptic longitude of the Sun
<i>L</i>	shadow radius at the observer's site
<i>L</i> ₀	mean longitude of the Moon
<i>m</i>	apparent magnitude
<i>m</i>	observer's distance from the shadow axis
<i>M</i>	mean anomaly
<i>M</i>	magnitude of a solar eclipse
<i>M</i> _h	mean anomaly (hyperbola)
<i>M</i> _☉	mass of the Sun
MJD	Modified Julian Date
<i>n</i>	triangle area ratio
<i>n</i>	daily motion
<i>p</i>	semi-latus rectum
<i>P</i>	position angle of the contact points of a solar eclipse
<i>P</i>	Gauss vector (in direction of perihelion)
<i>q</i>	perihelion distance
<i>Q</i>	Gauss vector (perpendicular to perihelion)
<i>r</i>	general: distance
<i>r</i>	general: position vector
<i>r</i>	change in distance
<i>R</i>	Earth-Sun distance
<i>R</i>	radius of a celestial body
<i>R</i>	reduction to the ecliptic
<i>R</i>	refraction
<i>R</i>	Gauss vector perpendicular to the orbital plane
<i>R</i>	vector of the geocentric position of the Sun
<i>S</i>	area of sector
<i>t</i>	general: time
<i>T</i>	general: time in Julian centuries since J2000

T	orbital period
T_0	time of perihelion passage
$T_n(x)$	Chebyshev polynomial of the n -th order
ΔT	Ephemeris Time – Universal Time difference
TDB	Barycentric Dynamic Time
TDT	Terrestrial Dynamic Time
u	argument of latitude
U	auxiliary variable for near-parabolic orbits
UT	Universal Time
UTC	Coordinated Universal Time
v	general: velocity vector
V	position angle of the contact points of a solar eclipse
$V(1,0)$	visual magnitude at unit distance
W	orientation of the prime meridian
x, y, z	general: Cartesian coordinates
y	state vector
X, Y	standard coordinates (astrometry)
z	auxiliary angle in description of precession
α	right ascension
α_0	right ascension of the rotation axis
β	ecliptic latitude
δ	declination
δ_0	declination of the rotation axis
Δ	geocentric distance
Δ	area of triangle
ε	obliquity of the ecliptic
$\Delta\varepsilon$	nutation in ecliptic latitude
ζ	auxiliary angle in description of precession
η	sector-triangle ratio
ϑ	general: spherical coordinate
ϑ	auxiliary angle in description of precession
ϑ	position angle
Θ	local sidereal time
Θ_0	Greenwich sidereal time
λ	ecliptic longitude
λ	geographical longitude (positive westwards)
λ	planetographic longitude (positive opposite to the rotation)
λ'	planetocentric longitude (positive towards east)
Λ	auxiliary angle in description of precession
ν	true anomaly
π	3.1415926...
π	equatorial horizontal parallax
π	auxiliary angle in description of precession

Π	auxiliary angle in description of precession
ϖ	longitude of perihelion
ρ	geocentric distance
τ	hour angle
τ	interval
φ	general: spherical coordinate
φ	geographical (planetographic) latitude
φ'	geocentric (planetocentric) latitude
$\Delta\psi$	nutation in ecliptic longitude
ω	argument of perihelion
Ω	longitude of ascending node

$\emptyset$	diameter
Υ	vernal equinox (First Point of Aries)
$\oplus$	Earth
$\odot$	Sun
cy	century
d	day
h	hours
m	minutes
m	magnitudes
r	revolutions
s	seconds
o	degree

Glossary

Aberration: The shift of the apparent position of a celestial body with respect to the geometric position, caused by the finite speed of light. (→stellar aberration; →light-time).

Altitude: In spherical astronomy, the angle between a celestial body and the →horizon, as measured by an observer at the surface of the Earth. The actual and the observed altitude of a body differ, because of →refraction, by up to 35'.

Apparent Coordinates: The coordinates of a celestial body that are required, for example, for use with setting circles on a telescope. Apparent coordinates refer to the actual orientation of Earth's axis, and include corrections for →precession and →nutation. In addition, →stellar aberration is taken into account, and, in the case of Solar-System bodies, →light-time.

Ascending Node: The point on an orbit at which the celestial body crosses the ecliptic from south to north.

Astrometric Coordinates: Coordinates to be compared with catalogued stellar positions. Astrometric coordinates generally relate to the mean →vernal equinox of a standard epoch (→J2000, →B1950), and take into account →light-time in the case of planets or comets.

Astrometry: The measurement of stellar positions using divided circles or photographic plates.

Astronomical Unit: A unit of length that is used to specify distances within the Solar System. One astronomical unit (AU) is approximately equal to the mean distance between the Earth and the Sun, and amounts to approximately 149.6 million km.

Azimuth: One of the coordinates in the →horizontal system. Azimuth is the angle, measured positive towards the west, between south and the direction in which an object is observed.

B1950: The beginning of the Besselian year 1950 ($B1950 = JD\ 2433282.423 = Jan\ 0^d 923, 1950$). A reference epoch that has been used for a very long time, but which has now been superseded by the standard epoch of →J2000.

Central Meridian: The meridian passing through the center of the apparent disk of a planet as seen from Earth..

Culmination: The moment at which a celestial body reaches its greatest (or least) altitude above the →horizon. Culmination occurs when a body crosses the →meridian.

Declination: The angle, at right-angles to the →equator, measured between the equator and a celestial body. Together with →right ascension, declination forms the equatorial system of coordinates. (→equinox)

Dynamical Time (TDB, TDT): A physical measure of time, such as that obtained by the use of atomic clocks, for example. Dynamical Time is, like →Ephemeris Time, a uniform measure of time, but which takes into account the relationship between time and space required by relativity theory. While Barycentric Dynamical Time (TDB) gives the time that would be measured by an observer at the Solar System's centre of gravity, Terrestrial Dynamical Time (TDT) gives the time that would be shown by a clock at the centre of

the Earth. Both definitions of time differ between themselves and from →Ephemeris Time by just a few milliseconds. In 1991 it was agreed that Terrestrial Dynamical Time (TDT) should just be called →Terrestrial Time (TT).

Ecliptic: The imaginary great circle, representing the intersection of the plane of the Earth's orbit with the celestial sphere. As seen from the Earth, the Sun appears to travel along the ecliptic in the course of a year. The ecliptic serves as reference plane for the ecliptic system of coordinates, which uses the coordinates ecliptic longitude and ecliptic latitude (→ecliptic coordinates).

Ecliptic Coordinates: The position of a body is established relative to the →ecliptic and the equinox of some reference epoch, using the coordinates longitude and latitude. Ecliptic longitude is measured from the →vernal equinox, positive towards the east. Ecliptic latitude is the angle, measured perpendicular to the ecliptic, between the body and the ecliptic.

Elongation: The angle between two bodies, as seen by an observer.

Ephemeris: A table giving the coordinates of a planet, comet or minor planet for a specific period of time.

Ephemeris Time (ET): A uniform measure of time, which is used to calculate the coordinates of a planet, comet, or minor planet. Ephemeris Time was introduced to be independent of the irregular, unpredictable variations in the rotation of the Earth, which forms the basis for measuring →Universal Time. The difference between Universal Time (UT) and Ephemeris Time (ET) was, by definition, zero at the beginning of the 20th century, and now amounts to about one minute. Within the framework of relativistic theories of motion, Ephemeris Time has been replaced by →Terrestrial Time (TT).

Equator: An imaginary great circle on the celestial sphere, which is perpendicular to the Earth's axis of rotation. The equator separates the northern and southern celestial hemispheres, and is simultaneously the reference plane for the equatorial system of coordinates, which uses the coordinates →right ascension and →declination.

Equatorial Coordinates: Coordinates referred to the equator (→right ascension, →declination).

Geocentric Coordinates: Coordinates referred to the centre of the Earth.

Geographic Coordinates: Two values (geographic longitude and geographic latitude), which specify the position of a point on the Earth's surface. The reference plane is the terrestrial equator, and longitude is, by international agreement, measured from the Greenwich →meridian.

Heliocentric Coordinates: Coordinates referred to the centre of the Sun.

Heliographic Coordinates: Coordinates for uniquely describing a point on the surface of the Sun, defined by analogy with →geographic and →planetographic coordinates.

Horizon: The imaginary line of intersection between a plane tangent to the surface of the Earth at the observer and the celestial sphere. The horizon is the reference plane for the →horizontal system with coordinates →azimuth and →altitude.

Horizontal System: A coordinate system related to the local horizon of an observer, and where the coordinates used are →azimuth and →altitude. The horizontal system is used, for example, in calculating the rising and setting times of celestial bodies. It is, however, not suitable for ephemerides in an almanac, because the azimuth and altitude of a body depend on the observing site, and constantly alter because of the rotation of the Earth.

Hour Angle: Difference between the local →sidereal time and the →right ascension of a star. The hour angle measures the sidereal time that has passed since the last →culmination.

J2000: A standard epoch, which is widely used in astronomy. It represents midday on 2000 January 1 (2000 Jan. 1.5 = JD 2451545.0). The letter 'J' indicates that it is a Julian standard epoch. Successive Julian epochs normally differ by a full Julian century, consisting of 36525 days (e.g. J1900 = JD 2415020.0). (→B1950)

Kepler problem: Another description for the →two-body problem.

Light-Time During the time that light requires to cross the distance to the Earth (499 seconds per astronomical unit), a planet moves slightly farther along its orbit. The observed position therefore does not correspond to the actual point at which the planet was located at the time of observation.

Many-Body Problem: The task of calculating the motion of more than two bodies, taking into account their mutual gravitational attraction. Whereas the two-body allows for a closed-form solution, this is not generally the case with the many-body problem. Numerical methods are used to treat it, together with analytical approximations (series expansions).

Meridian: In astronomy, this is used to designate the great circle on the celestial sphere that passes through the observer's →zenith and intersects the →horizon at the north and south points (→culmination). In geography, the term 'meridian' means a great circle on the surface of the Earth that lies in the plane of the Earth's axis. All meridians pass through both of the Earth's poles and intersect the equator at a right angle. Meridians serve to define the geographical longitude of a point on the Earth.

Nutation: An oscillation of the Earth's axis about its mean position, which is superimposed on →precession. The period of nutation is 18.6 years, and is determined by the period for one rotation of the Moon's ascending node.

Parallax: The apparent alteration in the position of a celestial body relative to the background of fixed stars, and caused by changes in the observer's position. Parallax is greatest for the nearest celestial body, and is therefore most noticeable with the Moon and (to a far lesser extent) with comets and minor planets that pass close to the Earth (→topocentric coordinates).

Perigee: The closest point of the Moon's or a satellite's orbit to the Earth.

Perihelion: The closest point of a planetary or cometary orbit to the Sun.

Physical Ephemerides: Quantities describing the appearance of a planet as seen through the telescope, including e.g. the apparent diameter, the brightness, the location of the →central meridian and the →position angle of the rotation axis.

Planetographic Coordinates: Angles describing the location of a point on the surface of a planet, corresponding to →geographic coordinates on Earth.

Position Angle: The apparent angle on the celestial sphere between a reference direction and a given direction (e.g. pointing towards the Sun or the pole of a planet). Usually, the position angle is measured in degrees from North to East.

Prime Meridian: The reference meridian for determining longitude in the system of →planetographic coordinates.

Precession: The long-term shift in the positions of the ecliptic and the celestial equator. Because of perturbing forces exerted by the Sun and the Moon, the Earth's axis is not stationary in space. It sweeps out the circumference of a cone, centred on the mean pole of the ecliptic, with a period of 26000 years. As a result, the orientation of the celestial equator also changes. The forces exerted by the other planets also lead to a slow shift in the position of the ecliptic. The motion of the equator, →ecliptic, and →vernal equinox means that in giving the →equatorial and →ecliptic coordinates of a body the reference date must always be specified. Commonly the coordinates are referred to the equinox of date, the equinox of →B1950, or the equinox of →J2000.

Refraction: The change in the direction of rays of light in the Earth's atmosphere. A celestial body appears at a slightly greater altitude to an observer on the ground than it would without the effect of refraction. Close to the horizon, a ray of light has to pass through a particularly long portion of the atmosphere. Refraction is therefore most noticeable when a celestial body is rising or setting.

Right Ascension: One of the coordinates in the $\rightarrow$ equatorial system of coordinates. Right ascension is measured positive eastwards along the $\rightarrow$ equator from the $\rightarrow$ vernal equinox. The resulting angle is normally expressed as time ($15^\circ \equiv 1^h$), subdivided into hours, minutes and seconds.

Semi-diurnal Arc: Half of a star's apparent arc above the $\rightarrow$ horizon. When measured in hours, the semi-diurnal arc gives the $\rightarrow$ sidereal time between rising and $\rightarrow$ culmination and between culmination and setting.

Sidereal Time: The right ascension of a star that appears exactly on the observer's meridian. One sidereal day corresponds to the time taken for the Earth to rotate once, and lasts approximately 23^h56^m Universal Time.

Stellar Aberration: The difference between the direction of a ray of light as seen by an observer stationary with respect to the Sun, and one moving with the Earth.

Terrestrial Time (TT): Terrestrial Time is the relativistic timescale that nowadays replaces $\rightarrow$ Ephemeris Time (ET) as the time reference for apparent geocentric ephemerides ($\rightarrow$ Dynamical Time). For practical purposes both time scales may be considered to be equivalent.

Topocentric Coordinates: Coordinates related to the position of the observer on the surface of the Earth. Topocentric and $\rightarrow$ geographic coordinates differ by the amount of $\rightarrow$ parallax.

Two-Body Problem: The task of calculating the motion of two bodies under the influence of their mutual gravitational attraction. The two-body problem is a simplified representation of the motion of a planet, comet, or minor planet around the Sun, where perturbations caused by other bodies in the Solar System are neglected. The importance of the two-body problem for the calculation of ephemerides is that it is relatively simple to solve mathematically.

Universal Time: A measurement of time related to the rotation of the Earth, which forms the basis for civil time. Universal Time may be determined by the observation of the meridian transit of celestial bodies of known $\rightarrow$ right ascension. Because the rotation of the Earth is subject to irregular variations, Universal Time is not a uniform time scale ($\rightarrow$ Ephemeris Time, $\rightarrow$ Dynamical Time).

Vernal Equinox (First Point of Aries): The point of intersection of the $\rightarrow$ ecliptic and the $\rightarrow$ equator, at which the Sun crosses the equator from south to north, during its yearly passage along the ecliptic. This currently occurs about March 21 each year. The exact instant of this event defines the beginning of spring. The vernal equinox provides the reference point for the measurement of ecliptic longitude ($\rightarrow$ ecliptic coordinates) and $\rightarrow$ right ascension. Because the positions of the ecliptic and equator alter over the course of time through the effects of $\rightarrow$ precession, the date to which the coordinates refer must always be stated. Most frequently the equinox of date, the equinox of $\rightarrow$ B1950, and the equinox of $\rightarrow$ J2000 are employed.

Zenith: The imaginary point on the celestial sphere directly above the observer. The $\rightarrow$ altitude of the zenith therefore amounts to 90° .

Bibliography

The following listing contains a selection of basic and more comprehensive works on spherical astronomy and celestial mechanics, together with various items on mathematics and programming. They should serve to help the reader explore various individual points more deeply which cannot be treated in detail here.

General Works

- [1] *Astronomical Almanac*; U. S. Government Printing Office, Her Majesty's Stationery Office; Washington, London.
The leading English-language almanac. The 1984 edition contains a summary of the numerical values and constants (sidereal time, precession and nutation, planetary masses and diameters, elements for physical ephemerides, etc.), that had just been recommended by the IAU for astronomical calculations.
- [2] H. Buerius, M. Schneider; *Himmelsmechanik I-II*; Bd. 143/144, Bibliographisches Institut; Mannheim (1966).
A modern description of celestial mechanics. Among other subjects discussed are general perturbation theory, lunar theory, and orbital determination.
- [3] L. E. Doggett, G. H. Kaplan, P. K. Seidelmann; *Almanac for Computers for the Year 19xx*; Nautical Almanac Office, United States Naval Observatory; Washington.
This yearly publication contains descriptions of the coordinates of the Sun, Moon and planets in terms of Chebyshev polynomials, which allow simple and accurate calculation of any required ephemerides, even on small computers. The introductory chapter gives advice, and simple formulae for calculation of astronomical phenomena.
- [4] *Explanatory Supplement to the American Ephemeris and Nautical Almanac*; U. S. Government Printing Office, Her Majesty's Stationery Office; Washington, London (1974). Completely revised edition by P. K. Seidelmann (ed.); University Science Books (1992).
The volume supplements the yearly *Astronomical Almanac*. Details of the data used and calculation methods.
- [5] R. M. Green; *Spherical Astronomy*; Cambridge University Press; Cambridge (1985).
The modern standard work on spherical astronomy.
- [6] R. Herschel, F. Pieper; *Pascal und Pascal-Systeme*; Oldenbourg Verlag; München, 5. Aufl. (1985).
Textbook on the Pascal language.
- [7] D. McNally; *Positional Astronomy*; Muller Educational; London (1974).
Introduction to spherical astronomy.
- [8] J. Meeus; *Astronomical Algorithms*; Willmann-Bell; Richmond, Virginia (1991).

Revised and greatly extended edition of *Astronomical Formulae for Calculators*. Practically orientated collection of interesting and unusual calculation procedures for amateur astronomers. Comprehensive examples, which also can be implemented on small computers. In addition floppy disks with Basic, Pascal or C programs are available.

- [9] J. Meeus; *Astronomical Tables of the Sun, Moon and Planets*; Willmann-Bell; Richmond, Virginia (1983).
Comprehensive collection of interesting astronomical data and results include: phases of the Moon, solar and lunar eclipses, planetary conjunctions, occultations of planets by the Moon, etc.
- [10] I. I. Mueller; *Spherical and practical astronomy*; Frederick Ungar Publishing Co.; New York (1969).
Description of spherical astronomy and its applications to geodesy. Among other topics covered are: the basis for the various definitions of time and their practical application; and the prediction and reduction of solar eclipses and lunar occultations.
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery; *Numerical Recipes*; Cambridge University Press; Cambridge, 2nd ed. (1992).
A comprehensive collection of important procedures covering all fields of mathematical computation. Converted into Fortran, Pascal or C-programs, which are also available on diskette.
- [12] A. E. Roy; *Orbital Motion*; Adam Hilger Ltd.; Bristol, 2nd ed. (1982).
General introduction to celestial mechanics.
- [13] H. R. Schwarz; *Numerische Mathematik*; B. G. Teubner Verlag; Stuttgart, 2. Aufl. (1988).
Practically orientated description of important algorithms in mathematical computation.
- [14] K. Stumpff; *Himmelsmechanik I-III*; VEB Deutscher Verlag der Wissenschaften; Berlin (1959,1965,1974).
Comprehensive discussion of celestial mechanics.
- [15] R. S. Pressman; *Software Engineering - A Beginner's Guide*; McGraw-Hill, Inc. (1988).
Practically orientated discussion of modern software engineering techniques.

Chapter 2 (Coordinate Systems)

- [16] J. H. Lieske, T. Lederle, W. Fricke, B. Morando; *Expressions for the Precession Quantities Based upon the IAU (1976) System of Astronomical Constants*; Astronomy and Astrophysics, vol. 58, pp. 1-16 (1977).
A list of the various auxiliary angles required to describe precession in ecliptic and equatorial coordinates. The numerical values are based upon the IAU's 1976 system of astronomical constants.
- [17] G. Moyer; *The Origin of the Julian Day System*; Sky and Telescope, vol. 61, pp. 311-313 (April 1982).
Article on the history of the Julian Date, with formulae to convert between the running date number and the ordinary calendar.

Chapter 3 (Rising and Setting Times)

- [18] L. D. Schmadel, G. Zech; *Empirical transformations from UT to ET for the period 1800-1988*; *Astronomische Nachrichten*, vol. 309, pp. 219-221 (1988).

Representation of the observed difference between Universal Time and Ephemeris Time by a 12th-order polynomial, with validity over a period of almost 200 years.

Algorithms for the calculation of rising and setting of objects and for handling refraction are given in *Almanac for Computers* [3].

Chapter 4 (Cometary Orbits)

- [19] J. M. A. Danby, T. M. Burkhardt; *The Solution of Kepler's Equation I-II*; *Celestial Mechanics*, vol. 31, pp. 95-107, pp. 317-328 (1983);

Discussion of various iterative procedures for solving the Kepler equation and the corresponding starting values.

- [20] B. Marsden; *Catalogue of Cometary Orbits*; Smithsonian Astrophysical Observatory; Cambridge, Mass.;

Regularly revised listing of osculating elements for all known comets.

- [21] *Ephemerides of Minor Planets*; Institute for theoretical Astronomy; St. Petersburg.

Orbital elements and ephemerides for the known minor planets. Appears yearly.

Methods for handling near-parabolic orbits after Stumpff are described in *Himmelsmechanik I* [14].

Chapter 5 (Perturbations)

- [22] L. F. Shampine, M. K. Gordon; *Computer Solution of Ordinary Differential Equations*; Freeman and Comp., San Francisco (1975).

Introduction to the theory of multistep methods for the numerical solution of ordinary differential equations and description of the variable order variable step size Adams method DE.

The most recent orbital elements of minor planets are published annually in the *Ephemerides of Minor Planets* [21] of the St. Petersburg Institute of Theoretical Astronomy as well as in various astronomical yearbooks.

Chapter 6 (Planetary Orbits)

- [23] P. Bretagnon, J.-L. Simon; *Tables for the motion of the sun and the five bright planets from -4000 to +2800; Tables for the motion of Uranus and Neptune from +1600 to +2800*; Willmann-Bell; Richmond, Virginia (1986).

Series expansions and programs, with which the motion of the inner planets during historical times can be calculated with sufficient accuracy. The coordinates of the outer planets are represented by polynomials.

- [24] T. C. van Flandern, K. F. Pulkkinen; *Low precision formulae for planetary positions*; *Astrophysical Journal Supplement Series*, vol. 41, p. 391; (1979).

Series expansions for the calculation of low-accuracy (ca. 1') heliocentric and geocentric planetary coordinates.

- [25] E. Goffin, J. Meens, C. Steyaert; *An accurate representation of the motion of Pluto*; Astronomy and Astrophysics, vol. 155, p. 323; (1986).
Representation of Pluto's orbit by a trigonometric series, which was obtained by fitting a numerically integrated ephemeris.
- [26] G. W. Hill; *Tables of Jupiter, Tables of Saturn*; Astronomical Papers of the American Ephemeris, vol. VII, part 1-2; Washington (1898).
Analytical series expansions of the motions of Jupiter and Saturn.
- [27] M. P. Jarnagin, jr.; *Expansions in elliptic motion*; Astronomical Papers of the American Ephemeris, vol. XVIII; Washington (1965).
Series expansions for the equation of the centre and radius, as well as other coordinates in the non-perturbed Kepler problem to high orders of eccentricity.
- [28] S. Newcomb; *Tables of the motion of the Earth, Tables of the heliocentric motion of Mercury, Tables of the heliocentric motion of Venus, Tables of the heliocentric motion of Mars*; Astronomical Papers of the American Ephemeris, vol. VI, part 1-4; Washington (1898).
Analytical series expansions for the motions of the inner planets.
- [29] S. Newcomb; *Tables of the heliocentric motion of Uranus, Tables of the heliocentric motion of Neptune*; Astronomical Papers of the American Ephemeris, vol. VII, part 3-4; Washington (1898).
Series expansions of Uranus and Neptune coordinates
- [30] X. X. Newhall, E. M. Standish Jr., J. G. Williams; *DE102: a numerically integrated ephemeris of the moon and planets spanning forty-four centuries*; Astronomy and Astrophysics, vol. 125, pp. 150-167 (1983);
The description of what is currently the best ephemeris over a period of several thousand years.
- [31] F. E. Ross; *New Elements of Mars*; Astronomical Papers of the American Ephemeris, vol. IX, part 2; Washington (1917).
Improvement of the orbital elements in Newcomb's tables of the martian orbit.

Chapter 7 (Physical Ephemerides)

- [32] M. E. Davies, V. K. Abalakin, J. H. Lieske, P. K. Seidelmann, A. T. Sinclair, A. M. Sinszi, B. A. Smith, Y. S. Tjuffin; *Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1982*; Celestial Mechanics, vol. 29, pp. 309-321 (1983).
Describes the rotational elements of the major planets used in the *Astronomical Almanac* [1] since 1984. Recent results of the working group on cartographic coordinates of International Astronomical Union are published approximately every three years. (Celestial Mechanics vol. 22, pp. 205-230 (1980), vol. 39, pp. 103-113 (1986), vol. 46, pp. 187-204 (1989), vol. 53, pp. 377-397 (1992)).

A detailed description of the computation of physical ephemerides is furthermore given in the *Explanatory Supplement* [4].

Chapter 8 (The Orbit of the Moon)

- [33] E. W. Brown; *An introductory treatise on the Lunar Theory*; Cambridge University Press (1896), Dover Publications (1960).

Description of the various perturbation-theory methods of analytically handling the motion of the Moon.

- [34] M. Chapront-Touzé, J. Chapront; *ELP 2000-85: a semi-analytical lunar ephemeris adequate for historical times*; Astronomy and Astrophysics, vol. 190, p. 342; (1988).
Mean orbital elements and perturbation terms describing the orbit of the Moon over long periods of time.
- [35] M. C. Gutzwiller, D. S. Schmidt; *The motion of the moon as computed by the method of Hill, Brown and Eckert*; Astronomical Papers of the American Ephemeris, vol. XXIII, part 1; Washington (1986).
Analytical series expansions of the main problem relating to the motion of the Moon (without taking planetary perturbations into account).
- [36] *Improved Lunar Ephemeris 1952-1959*; Nautical Almanac Office; Washington, 1954.
Revised version of Brown's lunar theory of 1954 with all necessary perturbation terms and mean arguments.

A good introduction to the theory of the lunar orbit and the basic perturbations can be found in the volume *Himmelsmechanik II* by Bucerius and Schneider [2]. The method of approximating functions by Chebyshev polynomials is explained in textbooks on numerical mathematics (see [11] and [13]).

Chapter 9 (Solar Eclipses)

- [37] T. Oppolzer; *Canon der Finsternisse*; Denkschriften der Math.-Naturw. Classe der Kaiserlichen Akademie der Wissenschaften Bd. 52, Wien (1883); Dover Publications, Inc., New York (1962).
Elements for solar and lunar eclipses for the period -1207 to 2163. Maps of the paths of the centre line for total solar eclipses.
- [38] H. Mücke, J. Meeus; *Canon of Solar Eclipses, -2003 to 2526*; Astronomisches Büro, Hasenwartgasse 32, Wien (1984).
Elements for 10774 solar eclipses for about 4500 years on the basis of Newcomb's theory of the solar orbit, and Brown's lunar theory. Simple diagrammatic charts showing the approximate path of the centre line.

The book *Astronomical Algorithms* by J. Meeus [8] describes various methods, which are particularly suitable for rapid prediction and assessment of possible eclipse dates. Ready to use tables for all kind of solar eclipses are given in [9]. The classical method of calculating solar eclipses using Besselian elements is described in the *Explanatory Supplement* [4] and in Mueller [10].

Chapter 10 (Stellar Occultations)

- [39] J. Robertson; *Catalog of 3539 Zodiacal Stars for the Equinox 1950.0* Astronomical Papers of the American Ephemeris, vol. X, part 2; Washington (1917).
Reference catalogue for the prediction and assessment of stellar occultations. Contains stars near the ecliptic down to about ninth magnitude that may be occulted by the Moon.

The prediction of stellar occultations is described in various textbooks on spherical astronomy (for example, Green [5]) and in the *Explanatory Supplement* [4]. More detailed treatment of the orbit of the Moon and the rotation of the Earth is discussed in *Spherical and practical astronomy* [10].

Chapter 11 (Orbit Determination)

- [40] C. F. Gauss; *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections*; Dover Publications; New York;
English translation of *Theoria motus corporum coelestium*, in which Gauss describes his method of orbital determination. The book is mainly of historical interest, but it is less suitable as a textbook on orbital determination.
- [41] H. Buerius; *Bahnbestimmung als Randwertproblem I-V*; *Astronomische Nachrichten*, vol. 278, 280, 281, 282 (1950-1955);
A series of five papers, in which Gauss' simplified and full methods of orbital determination are examined, amongst other subjects. Numerous examples illustrate possible problems that may occur in determining orbits.
- [42] D. L. Boulet; *Methods of orbit determination for the micro computer*; Willmann-Bell; Richmond, Virginia (1991).
Principles and methods of ephemeris calculation and orbit determination, including special perturbations and the orbit determination methods of Laplace, Olbers and Gauss. The listed Basic programs are also available on floppy disk.

An up-to-date introduction to orbital determination is also given by the corresponding chapter in *Himmelsmechanik I* by Buerius and Schneider [2].

Chapter 12 (Astrometry)

- [43] *SAO Star Catalog*; Smithsonian Astrophysical Observatory; Cambridge, Massachusetts (1966);
Four-volume star catalogue with positions, proper motions and magnitudes of 258 997 stars. Epoch and equinox 1950.0. The accuracy of the positions is around 1".
- [44] W. Dieckvoss; *AGK3 Star Catalogue of Positions and Proper Motions North of -2°5 Declination*; Hamburg-Bergedorf (1975).
Comprehensive catalogue of positions for the northern hemisphere for astrometric work. Based on photographic surveys carried out by the Bonn and Hamburg-Bergedorf Observatories. Equinox 1950.0.
- [45] S. Röser, U. Bastian; *PPM Star Catalogue*; Spektrum Akademischer Verlag, Heidelberg (1991).
Successor of the SAO Catalog with positions and proper motions of approximately 360 000 stars of the northern and southern hemisphere referred to the equinox J2000 (FK5).
- [46] W. Tirion, B. Rappaport, G. Lovi; *Uranometria 2000.0*; Willmann-Bell; Richmond, Virginia (1987).
Atlas of the whole northern and southern skies with 473 charts at a scale of $1.85 \text{ cm} = 1^\circ$ for equinox 2000. Contains stars down to $9^m.5$ based on the Bonner Durchmusterung, the southern Bonner Durchmusterung, and the Cordoba Durchmusterung.

The calculation of standard coordinates is treated in many textbooks on spherical astronomy (cf. [5]). Details of regression fitting can be found in books on numerical mathematics, such as [11] and [13].

Subject Index

- ABERRAT 208
- Aberration 121,147,208
 - planetary 119
 - stellar 120
- ACCEL 86
- ACS 8
- Adams 91
- Addition theorems 114,158
- ADD SOL 159
- ADD THE 158,160
- Algorithm 4
- Altitude 35
 - geocentric 45
 - topocentric 45
- Angular measurements 10
- Anomaly
 - eccentric 62,236
 - mean 95,108,236
 - mean,daily variation 237
 - true 60,230,236
- APPARENT 218
- Apparent places 206
 - example 208
- Approximation
 - Chebyshev 167
 - error 169
 - Hansen 231
- ASN 8
- Astrometry, error estimation 261
- Astronomical Almanac 108
- Astronomical unit 63
- ATN 8
- ATN2 9
- Axis
 - semi-major 61,230,236
- Azimuth 35
- Barker's equation 64
- Bashforth 91
- Basic 7
- BESSEL 202
- Besselian year 17
- Brahe, Tycho 154
- BRIGHT 145
- Brightness, apparent 144
- Brown
 - E.W. 157
 - lunar theory 157
- Bucerius 229
- CALDAT 13
- Calender
 - Julian 12
 - reform 12
- Camera, axis of 254
- CART 10
- Celestial Pole 35,131
- Central meridian 136
- Centre
 - equation of the 108,154
- Centripetal force 60
- Century
 - Gregorian 15
 - Julian 15
- Charlier, boundary line 244
- Chebyshev approximation 167,169
- Chebyshev expansion 211
 - calculation of coefficients 169,170
 - evaluation 172
- Chebyshev polynomials 167
 - recursion 168,170
 - roots 168,170
- Circumpolar star 48
- Clenshaw, algorithmus of 172
- Clocks, atomic 42
- COCO 7,27
 - menu 30
- Coefficients
 - Chebyshev 169
 - Fourier 108
 - station 205,216,218
- COMET 59,78,245
- Comet Bradfield 253
- Conic section
 - equation of 60,235,236
- CONJUNCT 212,218

- Conjunction 206
 - iterative determination 210
- Contact times 199
- CONTACTS 202
- Coordinates
 - apparent 118,124
 - astrometric 77,118,120
 - Cartesian 9,9,36,75,208
 - derivatives 121
 - ecliptic 7,13,71
 - equatorial 7,13,14,184,245,257
 - fundamental plane 213
 - geocentric 7,23,45
 - geographic 135
 - geometric 118
 - heliocentric 7,23,75,76,117,118,131
 - lunar 170
 - mean 207
 - planetographic 135
 - polar 9
 - retarded 77
 - solar 23,76
 - spherical 229,255
 - standard 256,261
 - topocentric 45
 - true 207
- CROSS 237
- Cross-product 95,237
- CS 8
- CUBR 9
- Culmination 57
-
- DARKNESS 218
- Day
 - length of 40,42
 - sidereal 40
- DDD 11
- DE 91,93
- Declination 15,185
- Diameter
 - apparent 57,146
- Differential equation 89
- Distance 239,240
 - geometric 119
- DMS 11
- DOT 232
- Dot product 231,240
-
- Earth
 - axis of 13
 - flattening 184,214
 - mean pole 124
 - orbit perturbations 26
 - revolution of 120
 - rotation of 120
 - true pole 124
 - velocity of 208
- Earth-orbit solution 244
- ECCANOM 66
- Eccentricity 60,230,235
- ECLEQU 16
- ECLIPSE 191,192
- Eclipse
 - annular 181,202
 - central line 191
 - central phase 184
 - contact times 199
 - duration 191
 - local circumstances 199
 - magnitude 201
 - partial 184
 - position angle 199
 - solar 177,205
- Ecliptic 13,17,73,124,154,177,205,233
- obliquity of 15,124
- reduction to 110
- ECLTINER 202
- ELEMENT 237
- Elements
 - orbital 59,71,229,234,242
- ELLIP 66
- Ellipse 60,108,236
- Ellipsoid
 - rotational 184,214
- Elongation 143
- Ephemeris
 - Improved Lunar 157
 - lunar 173
 - physical 131
- Ephemeris Time (ET) 42,117,187,214
- Epoch 207,217
 - J1900 15
 - J2000 15,208
 - Julian 15
- EPSMACH 268
- Equation
 - Barker's 64
 - differential 89
 - Kepler's 63,109,236
 - of conic section 60,235,236
 - of motion 84
 - of orbital plane 239,240
 - of the centre 108,154
- Equator 13,124
 - celestial 206
- EQUECL 16
- EQUHOR 37

- example 41
- Equinox 7,17,72,118,217
 - B1950 17
 - J2000 17,147
 - mean of date 155
 - of date 17,117,124,207
 - vernal 13,72,73,124,206
- EQUSTD 256
- ETMINUT 187,214
- Evection 154
- EXAMINE 218
- F 93
- Film-plane 254
- FIND_ETA 232
- First Point of Aries 13
- Flattening 146
 - planetary 135
 - terrestrial 184,186,214
- Focal length 253
- Fortran 7,92
- FOTINP 261
- FOTINP, example 261
- FOTO 262
- Fourier series 108
- Functions
 - hyperbolic 64
 - Stumpff 70
 - trigonometric 7,158
- Fundamental plane 184,199
 - coordinates 213
- Gauss, C.F. 229
- Gaussian vector 71,74,234
- GAUSVEC 74
- GEOCEN 122
- GETINP 261
- Givens rotations 259
- Goffin, E. 108
- Gordon 92
- Gravitational constant 84
 - Gaussian 63
- Greenwich, meridian of 186
- Hansen approximation 231
- HOREQU 37
- Hour angle 36,40,41
- HYPANOM 67
- HYPERB 68
- Hyperbola 60
- ILLUM 143
- Illumination angle 144
- Improved Lunar Ephemeris 157

- Inclination 71,233
- Inequality, annual 154
- Integration, numerical 89
- Interpolation
 - Lagrangian 167
 - polynomial 167
 - quadratic 49,215
- Interval 235,241
- INTRP 92
- INTSECT 191
- Invariable plane 131
- Julian Date 11,117
- JUP200 115,118
- KEPLER 75
- Kepler, Johannes 60
- Kepler's equation 63,109,236
 - hyperbolic orbits 64,67
 - iteration 65
 - solution 65
- Kepler's laws
 - first 59
 - second 59,230
 - third 63,237
- Lagrange, J.L. 167
- Latitude
 - argument of 156,234
 - ecliptic 15
 - geocentric 185,214
 - geographic 35,185,214,216
 - planetocentric 135
 - planetographic 135
- Law
 - Kepler's first 59
 - Kepler's second 59,230
 - Kepler's third 63,237
 - of equal areas 61,230
 - of gravity 84
- Leap seconds 43
- Least-squares, method of 257
- Light
 - velocity of 76,120,208,243
- Light-time 76,76,118,119,121,190,243
- Line of nodes 156
- LMST 41,186,214
- Local circumstances 199
- Longitude
 - ecliptic 15,177
 - geographic 185,186,216
 - heliographic 136,147
 - of node 72
 - planetocentric 136

- planetographic 136
- LSQFIT 259,260
- LUNA 173
- Lunar theory, mean arguments 157
- Machine accuracy 93,268
- Magnitudes 144
- MAR200 118,278
- Mayer
 - lunar tables by 154
 - Tobias 154
- Meeus, J. 108
- MER200 118,273
- Meridian 35,256
 - central 136
 - of Greenwich 186
 - prime 137
- MINI_MOON 38,156
- MINI_SUN 39
- MJD 12
- Modified Julian Date 12
- Month
 - anomalistic 154
 - synodic 154
- MOON 38,157,161,211
- Moon
 - altitude 43
 - ascending node 154
 - attraction by the Earth 153
 - attraction by the Sun 153
 - Brown's theory 190
 - declination 166
 - ecliptic longitude 155
 - horizontal parallax 161
 - inclination of orbit 154,156,177,210
 - Keplerian orbit 153
 - latitude 178
 - mean anomaly 155
 - mean distance from node 155
 - mean elongation 155
 - mean longitude 155
 - mean motion 154
 - mean orbit of 154
 - orbit of 205
 - perturbation by Sun 153
 - phases of 154
 - planetary perturbations of 160
 - precession 154
 - right ascension 166
 - secular perturbations 154
 - series expansion 170
 - shadow of 205
- MOONEQU 166,211
- Motion
 - equation of 84
 - Keplerian 108,113
- Multistep method 90
- NEP200 118,288
- New Moon 177
 - date of 177,191
- NEWMOON 179
- Newton
 - Isaac 60,154
- Newton's method 65,167
- Node
 - line of 156
 - longitude of 234
- NORM 232
- Numerical integration 89
- NUMINT 99
- Nutation 124,147,190,207
 - in longitude 124
- NUTEQU 125
- Objective 253
- OBSERVER 202
- OCCULT 205,218
- Occultation
 - stellar 44,167,205
- ORBDAT 245,245
- ORBECL 75
- ORBINF.DAT, example 245
- Orbit
 - cometary 59
 - lunar 153
 - near-parabolic 68
 - non-periodic 60
 - periodic 60
 - planetary 107
 - trigonometric series 110
 - undisturbed Keplerian 107
- Orbit determination 229
 - comprehensive Gauss method 244
 - Gauss's method 229
 - Laplace method 229
- Orbital elements 59,229,234,242
 - minor planet Ceres 104
 - of the planets 87
 - osculating 95
- Orbital parameter 60
- Orbital period 236
- Orbital plane
 - equation of 239,240
- ORIENT 137
- PARAB 71

- Parabola 60
- Parallactic angle 201
- Parallax 45
 - horizontal 45,57
- Pascal 7,92,233
- Penumbra 181,205
 - diameter 182,191
- Perihelion 60
 - argument of 72,236
 - distance 236
 - longitude of 72,236
 - time of passage 236
- Period, orbital 236
- Perturbation 83
 - acceleration 85
 - periodic 113
 - terms 108,111
- Perturbations, periodic 107
- Phase 143
- Phase angle 142
- Photography 253
- PHYS 147,148
- Planets
 - coordinates 87
 - equatorial radius 135
 - mass 85
- PLANPOS 126
- PLANRISE 57
- Plate constants 257,261
- Pleiades 206,217
- PLU200 118,290
- PMATECL 20
- PMATEQU 20,207
- PW_MATRIX 207,207
 - example 207
- POLAR 10,15
- Pole
 - celestial 131
 - north 131
- Polynomial, Chebyshev 167
- POSANG 134,144
- POSITION 88,147
- Position
 - geocentric 239
 - geometrical 76
 - heliocentric 239
- Position angle 199,215,218
 - of the rotation axis 132
 - of the Sun 144
- POS_ANGLES 202
- PRECART 21,207
- Precession 7,17,72,124,147,155,206
 - angles 19
 - example calculation 21
- Prime meridian 137
- Proper motion 206
- QUAD 202,215
- Radius, apparent 47
- Reference stars 253
- Refraction
 - atmospheric 46
 - law of 46
- Regula falsi 210
- Relativity
 - theory of 42,120
- Residuals 258
- Right ascension 15,185,186,206
- ROTATION 141
- Rotation
 - direct 132
 - orientation of axis 132
 - retrograde 132
 - systems of 136
- SAT200 118,282
- Saturn, rings 145
- Scaliger
 - Joseph Justus 12
 - Julius 12
- Secant procedure 231
- Sector, area of 230
- Sector-triangle ratio 229
 - iteration 231
- Sectors, area of 241
- Semi-diurnal arc 48
- Semi-latus rectum 60,230,235
- Semi-major axis 61,230,236
- SHADOW 215,218
- SHADOW_DIST 202
- Shampine 92
- SHAPE 135
- Sidereal time 40,40,48,57,131,186
 - of Greenwich 40
- SITE 214
- SN 7
- Solar time 40
- SOLAR1 159
- SOLAR2 159
- SOLAR3 159
- SOLARN 159
- State vector 89
- Station coefficients 205,216,218
- STDEQU 256
- STEP 92
- Step size 91

Steyart, C. 108

STUMPF 70

Sun

– mean anomaly 155

– mean motion 154

– observed position 190

SUN200 23,38,118

SUNEQU 190

SUNSET 44,51

TERM 159

Terminator 144

Time 41

– barycentric dynamic 42

– difference between ET & UT 43,187

– dynamic 41,43

– ephemeris 42,187,214

– observation of ET-UT 226

– sidereal 42

– solar 42,48

– terrestrial dynamic 42

– unit of 36,48

– universal 41,42,187,214

Time zone 43

TN 8

Totality, duration of 188

Triangle

– area of 230,235

Twilight 47

– civil 218

Two-body problem 131,230

T_EVAL 172,189,190,211

T_FIT_LBR 171,172,189,211

T_FIT_MOON 189,211

– example 189,211

T_FIT_SUN 191

Umbra 177,181,205

– diameter 181,188,191

Universal Time (UT) 41,117,187,214

Universal Time Coordinated (UTC) 43

URA200 118,285

Variation 154

Vector

– Gaussian 71,74,234

– magnitude 231

Velocity

– of Earth 208

– planetary orbit 121

VEN200 118,275

XYZKEP 97

Year, length of 42

Zenith 35

Zodiacal Catalogue 217